

# Tabletop RPG character creator website

---

Bram Williams

# Contents

---

<b>Definition .....</b>	<b>8</b>
Business and end user.....	8
<b>Investigation and Analysis.....</b>	<b>9</b>
E-mails.....	9
Interview .....	10
aims.....	10
plan .....	10
Interview transcript.....	11
Analysis of Interview .....	13
Data flow diagrams .....	14
Log on.....	14
User home page .....	15
Admin home page .....	15
Choose race.....	16
Choose background .....	16
Determine ability scores .....	17
Choose a class .....	17
Character sheets .....	18
Main sheet .....	18
Secondary sheet.....	19
Spellcasting sheet.....	20
Alternative solutions .....	21
Using the free basic rules, or players handbook.....	21
D&D Character Builder.....	21
Dungeonscape.....	21
Requirements specification .....	22
User requirements .....	22
Hardware and software requirements .....	24
<b>Nature of the Solution.....</b>	<b>25</b>
Design specification .....	25
Justification .....	25

Interface design .....	26
Character creation .....	27
Race choice .....	27
Name and physical characteristics choice .....	28
Background choice.....	28
Ability score choice .....	30
Class choice .....	32
Character viewer.....	32
Character Leveller .....	34
<b>Algorithms.....</b>	<b>37</b>
Modular design .....	37
Data structures.....	38
Algorithms and pseudo code .....	39
Created character class.....	39
Race setter methods .....	41
Background Setter methods .....	44
Class Setter methods .....	46
Random ability score.....	49
Ability modifier .....	50
Random Heights and weights .....	50
Calculate skill modifiers .....	51
Calculate Armour Class .....	53
Login Database.....	55
login functions.....	55
Datastore character functions .....	58
Trace tables.....	60
Random ability score.....	60
Random Heights and weights .....	61
Trace table clarification.....	62
Test Strategy .....	63
<b>Software Development .....</b>	<b>66</b>
Making the main class.....	66
The Datastore.....	67
Character choice JSP pages.....	68

<b>Testing.....</b>	<b>73</b>
White box.....	73
Black Box testing .....	74
Main pages.....	74
Character sheet pages .....	82
Results of testing.....	88
User testing.....	90
<b>Evaluation .....</b>	<b>93</b>
Input requirements.....	93
Processing requirements .....	95
Design requirements.....	96
Output requirements.....	97
User response .....	98
Input success.....	99
Processing success .....	100
Design success.....	101
Output success.....	101
Final result:.....	101
User testing results .....	102
Evaluation of user response.....	104
Extensions .....	106
Extensions .....	106
Maintenance .....	108
Adaptive maintenance.....	108
Perfective maintenance .....	108
Corrective maintenance.....	108
<b>User Documentation.....</b>	<b>109</b>
Getting to the website .....	109
Creating an account .....	109
Logging in .....	110
Creating a character.....	110
Choosing your race .....	111
Choosing your name and appearance .....	113
Choosing your background .....	113

Choosing your alignment and personality .....	115
Choosing your characters back story .....	115
Choosing your ability scores .....	116
Choosing random ability scores .....	117
Choosing preset ability scores .....	118
Choosing point bough ability scores .....	118
Choosing your class.....	119
Choosing class options .....	120
Character confirmation.....	120
Viewing characters.....	121
Character sheets .....	121
Hardware and software requirements .....	122
Troubleshooting.....	123
I can't create an account.....	123
I can't login.....	123
My page keeps refreshing and not sending me to the next page .....	123
<b>Final code .....</b>	<b>125</b>
CC Package .....	125
Coursework.java .....	125
Processing.java.....	168
Helper package .....	170
Login.java .....	170
login.....	177
Index.jsp .....	177
account.jsp .....	179
Race selection .....	181
race.jsp .....	181
radio.js.....	191
valid.js .....	217
Background selection.....	218
nameAppearance.jsp .....	218
background.jsp.....	222
backgroundJ.js.....	228
acolyte.jsp .....	268

charlatan.jsp.....	272
criminal.jsp.....	276
entertainer.jsp .....	280
folkHero.jsp.....	284
guildArtisan.jsp .....	288
hermit.jsp.....	292
noble.jsp.....	296
outlander.jsp.....	300
sage.jsp.....	304
sailor.jsp .....	308
soldier.jsp .....	313
urchin.jsp.....	317
backstory.jsp .....	321
Ability score selection .....	323
abilChoice.jsp .....	323
randAbil.jsp .....	324
preset.jsp.....	329
pointBought.jsp.....	333
validRadio.js .....	339
Class selection.....	343
class.jsp .....	343
barbarian.jsp .....	350
bard.jsp .....	353
cleric.jsp .....	357
druid.jsp .....	368
fighter.jsp .....	371
monk.jsp.....	375
Paladin.jsp .....	378
ranger.jsp .....	381
rogue.jsp .....	385
rogueExpertise.jsp .....	388
sorcerer.jsp .....	390
warlock.jsp .....	395
wizard.jsp .....	398

finalise .....	400
sheet.jsp .....	400
Character viewer .....	403
characterViewer.jsp .....	403
print.jsp .....	405
print2.jsp .....	421
print3.jsp .....	423
courseworkApp.css .....	425
printDisplay.css .....	429
Appengine-web.xml .....	440
Web.xml .....	441
URL .....	442

## Definition

### Business and end user

Liam Murray is a student from the Grammar school Westcliff High School for Boys, he studies computing, biology and geology; with an aim to go into graphical design. He plays pen-and-paper RPGs and is a part of Westcliff High School for Boy's tabletop roleplaying games club as a hobby. As a part of running this club, the people he plays with need to create characters; play the core game and level up. Every Tuesday they play for an hour-long lunchtime slot.

A pen and paper role-playing games or RPG, is a game in which there is a group of players and a Gamemaster (GM). The players act out their character and describe their actions in the game either succeeding or failing depending on the formal rules of the RPG. The GM then describes the progression of the story and world, acting as a referee for the players. The title *pen and paper* is used to differentiate between this format and say a computer game RPG.

There are many different pen and paper RPG systems of rules to use, from the first 'Dungeons and Dragons' to 'Call of Cthulhu' but since D&D was created in 1974 by Gary Gyax and Dave Arneson, pen and paper RPGs have shared many of its gameplay mechanics and features:

- The use of 20 sided dice also called a 'D20' system to determine the outcome of actions.
- The participation of a GM, who creates the setting, represents all the non-playable characters (NPCs) and acts as a referee.
- The other players act as player characters (PCs) and the group as a whole is called the "party".

The party creates characters to portray in the game. Various systems have different character creation systems. Usually the player is able to choose:

- A race e.g., human, elf, dwarf.
- A class e.g., warrior, archer, wizard .
- Ability scores, these vary wildly from system to system.

Ability scores are very important as they are used in conjunction with the D20 to determine how successful someones actions were; this allows certain people to be better at doing certain things. This is almost always done on a physical paper medium, hence the name *pen and paper RPG*. The details are stored on a 'character sheet'. As the PCs play they will constantly refer to their sheets in order to find out how successful their rolls were; making them pivotal to play. Usually play results in PCs gaining experience and 'levelling up' this results in their character sheet being improved. The character sheet is in constant use.

This information was obtained through Wikipedia's tabletop roleplaying game page, as well as multiple YouTube videos on a variety of titles.

## Investigation and Analysis

Before proceeding, more needs to be found out about:

1. The end user
2. The system being used
3. The lunchtime club
4. What is needed

In order to investigate these points, an interview with Mr. Murray in addition to E-mails will be necessary.

### E-mails

Dear Liam

You were talking to me in school about your RPG club and I was just hoping you could tell me a bit about yourself and the club

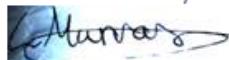
Kind regards Bram

Hello Bramwell 😊

Would love to help you there, is there anything in particular you would like me to cover? We play a role playing game in which we design a character and we hope a computer based program could speed up this process as it is very time consuming. We have 6 members including me, and we meet weekly ... if there is anything else you need to know for whatever reason do just ask 😊

Yours sincerely

Liam Murray



Hello Liam

can you describe the current system for creating characters please, what makes the process slow?

Could we meet for an interview; you could bring what's needed to create characters and we can talk about the system and what I can offer. When is best for you?

Kind regards

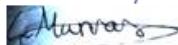
Bram

Hello,

The current system is to use a book for rules and fill out the details on paper, I can go for an interview tomorrow around 9:10am till 10am does that work for you? Using the book means lots of flicking through data and adding up values which I can show you tomorrow. I'll bring the book and the sheets we fill out, maybe if you made a character with me it would give you a helpful insight into the process needed

Sincerely

Liam Murray



Hello Liam

sure thing that'd be very helpful. See you then :)

Kind regards

Bram

## Interview

### aims

In the interview I have several primary aims for the information needed to create the solution.

- Information about the system being used, where the data that will be used can be found
- What kind of data will be involved
- What is needed of the program
- The environment the program is needed for
- Issues that need to be addressed by the system
- The clients requirements
- The audience of the program/ demographic
- Find out about existing solutions to the problem

### plan

#### Primary questions:

- What is involved in the club, what happens each session?
- Current system, what data is used and how (inputs and processes)
- What in simple terms is the issue with the current system?
- Set up base requirements specification (user? Hardware and software?)

#### Secondary questions:

- What other systems are available?
- What the audience for the software will be like
- How he wants the product distributed and accessed
- Figure out the best language( cross platform? online?)
- Get information on focus group of players

## Interview transcript

### Interview plan

Note: get pictures and information about the necessary books and the character sheet

- Players have book
- many systems
- ability to change system?

### Primary questions:

- What is involved in the club, what happens each session

Players meet in E6 on Tuesday lunch. The players get out their player sheets and dice. There is a recap of the last session. Then the GM (the player who runs the session sets the scene) will begin the game, asking players to make rolls when necessary. When characters play, they earn xp. This is used to level up characters that have been created (usually with help) over sessions.

- Current system, what data is used and how (inputs and processes)
- | Choices          | ability Scores | Personality traits | appearance                 | order creation                                   |
|------------------|----------------|--------------------|----------------------------|--|
| ability Scores   | ideals         | level up           | generate XP                | 1. race  |
| Saving throws    | bonds          | change level       | choose Proficiency bonuses | 2. class   |
| Skills           | flaws          |                    | • Hit points, Hit dice     | 3. generate ability scores (pick / random / buy) |
| Race             | Background     |                    | • Spells                   | 4. background, appearance, Personality           |
| Class (features) | Alignment      |                    | • Features                 | 5. equipment                                     |
| Hit Points       | Spells         |                    |                            |  |
| Items            |                |                    |                            |  |
- (They would like to create characters and level up in their spare time, but most players need help and the books; so use session time.)
- what in simple terms is the issue with the current system

I t takes a long, long time to create characters and level up, time which could be spent playing the game. Also paper character sheets need to be maintained and are easily ruined.

- Set up base requirements specification (user ? hardware and software?)

### User requirements

- Create characters fully
- level up characters
- keep a digital backup of characters  
optional to log onto, to access their personal data
- a way for the GM to see the player's character sheets
- the ability to print off useable information

### hardware / software requirements

- an account to log onto from any machine, that allows the user to access their personal data.
- simple options for creating the character (maybe a mode that recommends choices)
- tool tips explaining options.

Secondary questions:

- What other systems are available?
  - Dungeons & Dragons TBC
  - paper system and books

- swaps to another RPG, but they are fond of the ones they use

- what the audience for the software will be like
  - 6, Sixth Form students, male, 17-19, computer literate

- how he wants the product distributed and accessed

as an ~~external~~ online program For the players to access and update their characters

- Figure out the best language (cross platform? online?)

programming  
server  
side

- Java? - capable, OOP
- C++ - different but capable
- Python - simple
- VBA - optimized for forms

- Get information on focus group of players

talk to      name      character

Angus	-	Bard
Connor	-	dwarf

(other service?)

- google appspot for site
- online - HTML
- JavaScript
- CSS

Signature :



## **Analysis of Interview**

From speaking with the client, I found that in order to create a character, the players handbook was mandatory as it contained all the information for processing. The client lent me his players handbook so that I could begin designing the system, get a better grasp of the calculations the program would need to perform and create data flow diagrams until my players handbook arrives. It appears some aspects of the program will be very time consuming, a topic I brought up with the client and that will require further research. I also intend on speaking to the other members of the club for their thoughts on the project.

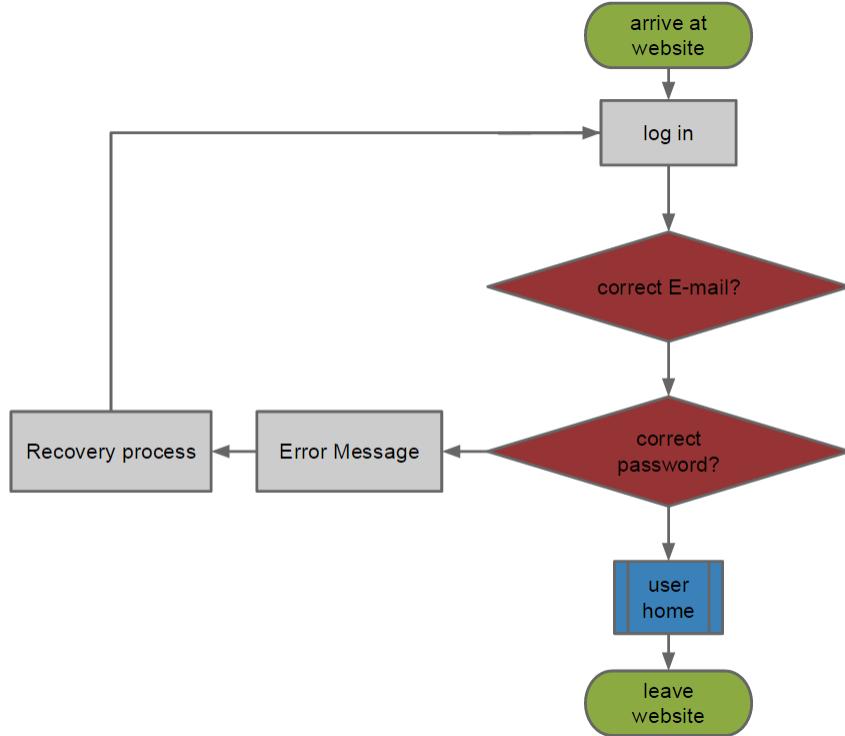
Mr. Murray insists that the product should be available as a browser based product so created characters can be accessed from at home and school, he also strongly wanted the website to be simplistic in layout, to be as easy to use as possible. Out of personal preference the client wanted the use of purple as the main theme of the website.

I feel optimistic about my abilities to create the product Mr. Murray wants and will move onto creating the data flow diagrams of the current system.

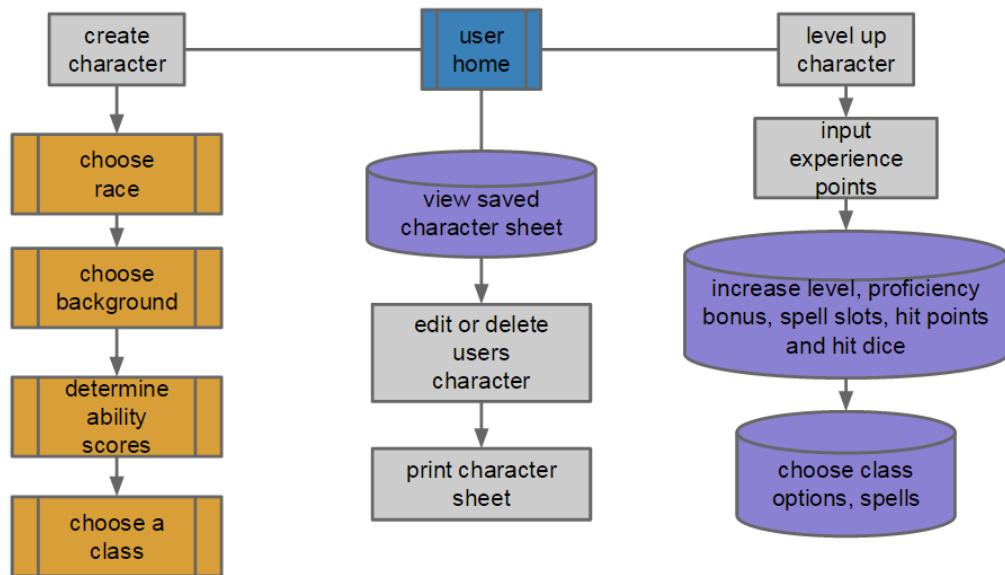
## Data flow diagrams

This information was all obtained from documentation, including the players handbook as well as the free online component of the rules that can be found here <http://dnd.wizards.com/articles/features/basicrules>. Using this information I could create data flow diagrams for the system.

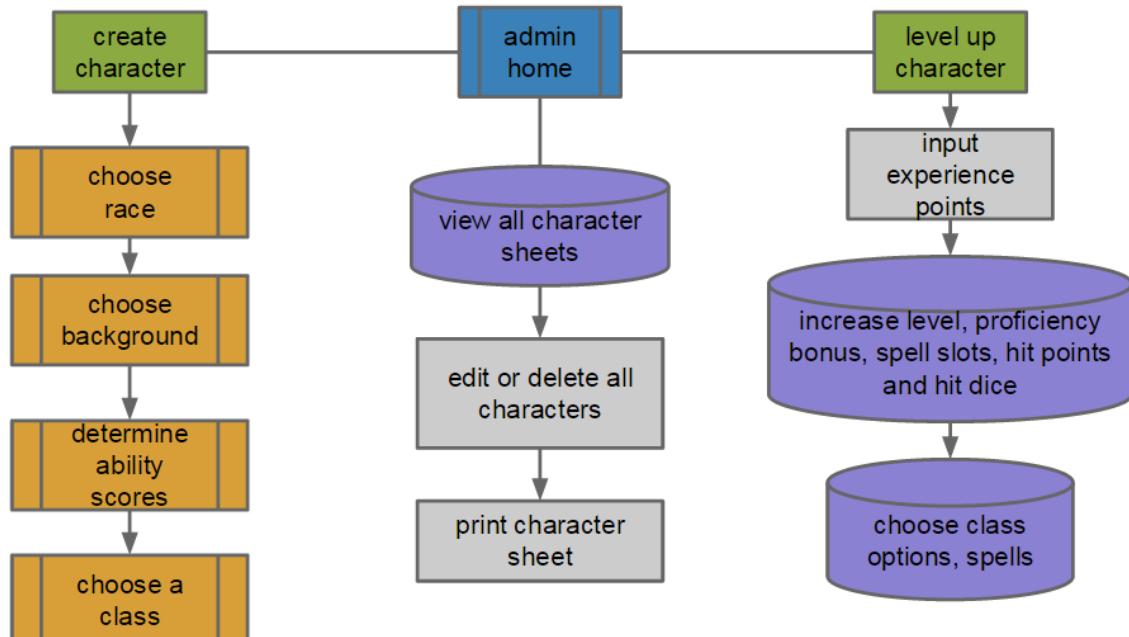
### Log on



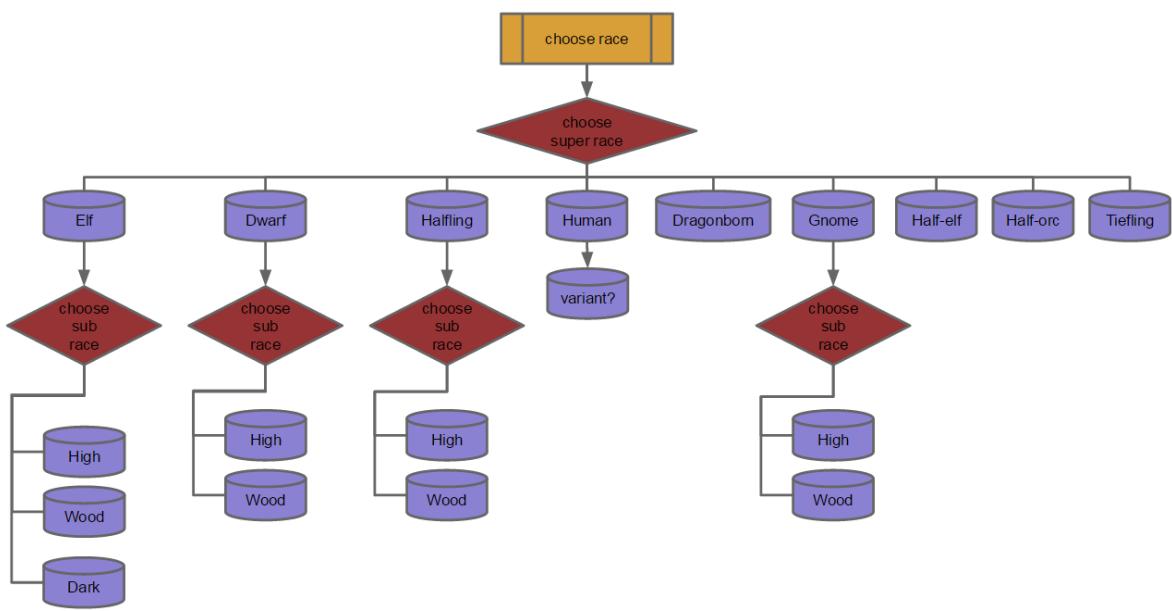
## User home page



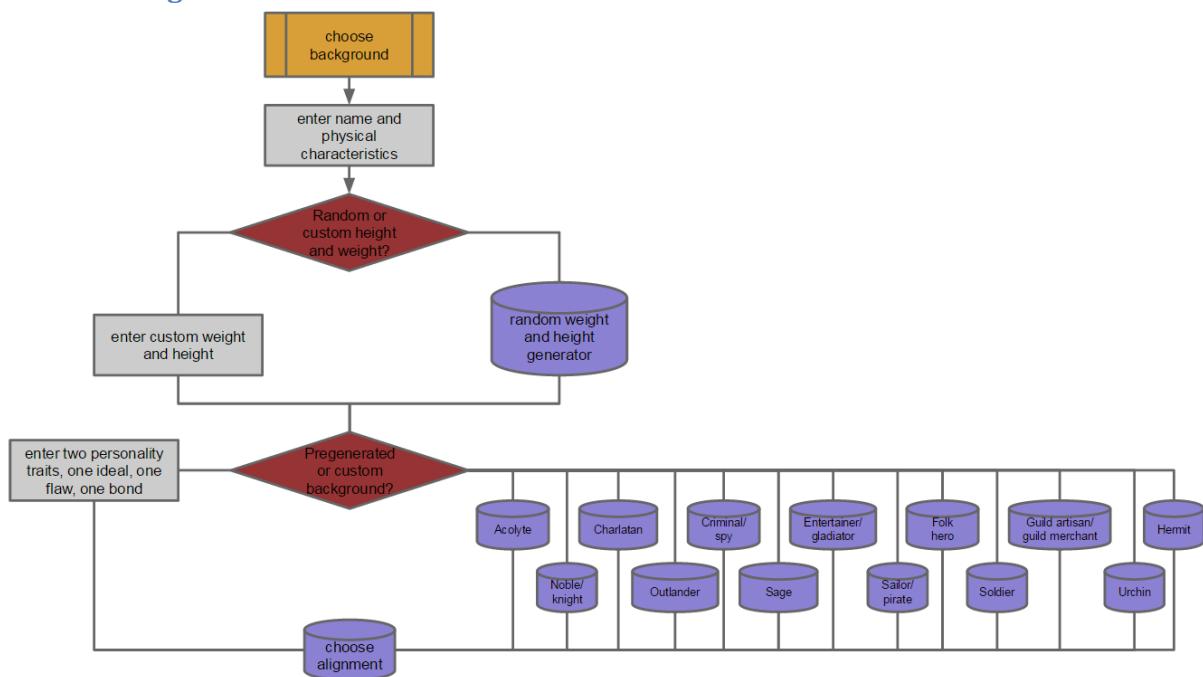
## Admin home page



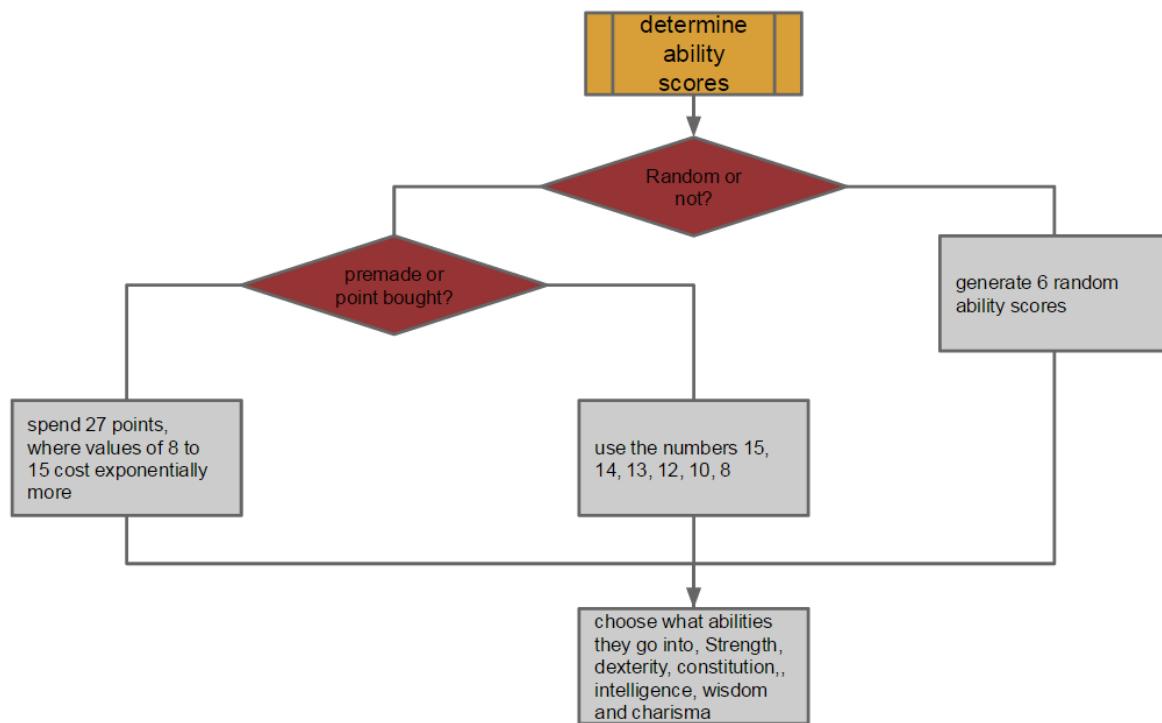
## Choose race



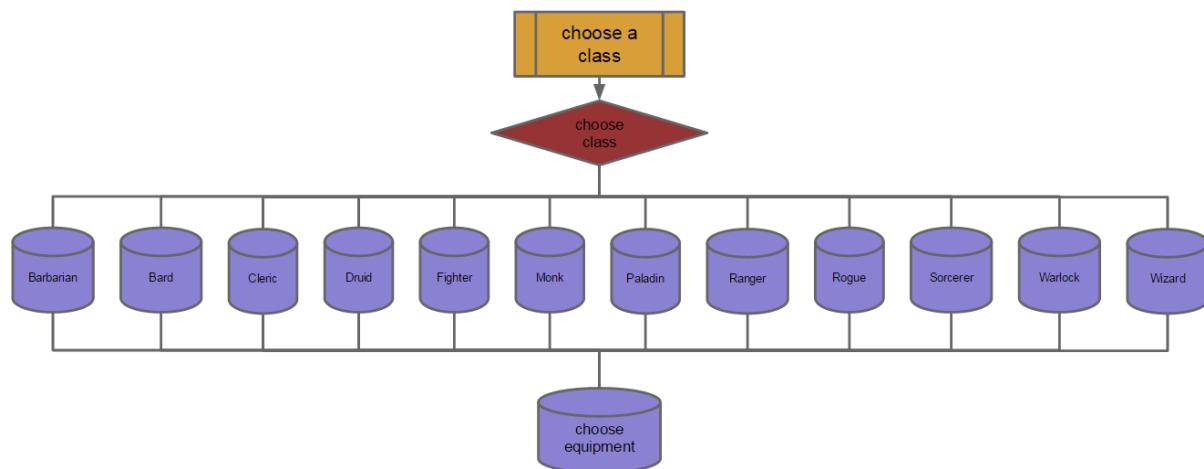
## Choose background



## Determine ability scores



## Choose a class



## Character sheets

The information will be put into character sheets like these, for use in games. These sheets can be found through the link to the basic rules.

## Main sheet

## Secondary sheet

The sheet is organized into several sections:

- Header:** Features the D&D logo with silhouettes of characters.
- Character Name:** A banner at the top left.
- Basic Stats:** Rows for Age, Height, Weight, Eyes, Skin, and Hair.
- Character Appearance:** A large section for drawing or describing the character's looks.
- Allies & Organizations:** A section with a grid for listing allies and their details.
- Additional Features & Traits:** A large section for listing special abilities or traits.
- Character Backstory:** A large section for writing the character's history.
- Treasure:** A section for listing the character's possessions.

TM & © 2014 Wizards of the Coast LLC. Permission is granted to photocopy this document for personal use.

## Spellcasting sheet

TM & © 2014 Wizards of the Coast LLC. Permission is granted to photocopy this document for personal use.

## Alternative solutions

### Using the free basic rules, or players handbook

The system the client uses currently involves either making a barebones character with the basic rules, available online, or by purchasing the \$50 handbook [https://dnd.wizards.com/products/tabletop-games/rpg-products/rpg\\_playershandbook](https://dnd.wizards.com/products/tabletop-games/rpg-products/rpg_playershandbook). However the client points out that this system is slowed down greatly by its format when creating characters. This is because the player needs to go back and forth constantly between chapters. Also the player needs to do a lot of arithmetic to determine lots of character scores that can go wrong very easily if the player is unfamiliar with the game. So although this is the standard way of making a character it could greatly be improved.

### D&D Character Builder

For systems older than the groups RPG that they play, Wizards of the Coast had an application that could be downloaded called the 'D&D Character Builder' <http://archive.wizards.com/dnd/tools.aspx>. This allowed users to create character for the fourth edition of 'Dungeons and Dragons', however this system has many limitations now. Firstly the user would need to play with the fourth edition rules which the client does not, also it is a very expensive product, that requires the user pays a from 6 to 10 dollars a month subscription and loses access when they stop playing. This is much too expensive seeing as campaigns of the game can last for many years with users wanting to keep making characters. As such this would be a very poor solution.

### Dungeonscape

Wizards of the coast was funding a character builder for the system my client uses, under the name 'Dungeonscape', formerly codename 'Morningstar'. This looked to be a useful product to be released in a few years, that allowed the user to create characters for the game as a phone and tablet application, however by the time the product would be released, the client would no longer be at the RPG club at his school. Also a few months ago funding for 'Dungeonscape' ceased and it is assumed that it is no longer under production.

[http://dnd.wizards.com/articles/news/digital\\_tools\\_announcement](http://dnd.wizards.com/articles/news/digital_tools_announcement)

# Requirements specification

## User requirements

1. Input requirements
  - a. Users should be able to log in to an account
  - b. Users should be able to create multiple characters
  - c. Users should be able to input all the character options:
    - i. Choose race
    - ii. Choose appearance
    - iii. Choose background
    - iv. Choose ability scores
    - v. Choose class
    - vi. Choose all subsequent choices within those listed above
  - d. Users should be able to choose created characters to view
  - e. Users should be able to increase their characters experience points
  - f. Users should be able to edit and delete created characters
  - g. Users should be able to use their mouse to select options
  - h. Users should be able to save their character sheets, to access whenever they need to
2. Processing requirements
  - a. The program should provide information about the choices the user makes
  - b. The program should recommend choices for classes
  - c. The program should be able to create characters and obtain all the appropriate information for choices
  - d. The program should allow Email and password recovery on login
  - e. The program should store created characters for future access
  - f. The program should be able to generate random values when wanted e.g. ability scores
  - g. The program should assimilate all the information in one easy to use document, if possible, a character sheet.
  - h. The program should calculate all values for the sheet including:
    - i. Ability scores
    - ii. Ability modifiers
    - iii. Saving throws
    - iv. Skill modifiers
    - v. Armour class
    - vi. Initiative
    - vii. Hit points
    - viii. Hit dice
    - ix. Passive perception
3. Design requirements
  - a. The layout should be minimalistic and easy to use
  - b. There should be a login screen first then a main menu with the three main options presented
  - c. When creating characters the screen should bring up further choices
  - d. Left to right, top down familiar design
  - e. Use of complementary colours
  - f. links to the main pages should be available on any part of the website in a bar at the top
  - g. Centralised design
  - h. Use of Pastel purple

#### 4. Output requirements

- a. Output all the information to the correct boxes in the character sheet
- b. The program should be able to output all three completed character sheets for the user to view
- c. The program should be able to print character sheets with all the needed information
- d. Output all created characters to the GM so they can see what the players characters are

After laying out the requirements specification I E-mailed the client a copy to check over and confirm they met the requirements he wanted.

Dear Liam,

I have put together the requirements specifications we discussed as attached. Can you confirm these are as you wanted and put forward any changes?

Kind regards,  
Bram

Hello Bram,

I am writing to give feed back on the requirements specification. Everything I want is on the file, there is nothing I want to change. Keep me posted.

yours faithfully,  
Liam Murray  
[sig.jpg](#)

The client had no changes he wanted to make, so I will finish the software and hardware requirements, then continue onto design.

## **Hardware and software requirements**

I intend to create the website using Google's Appengine. This will allow Google's cloud Datastore to be used as a database, and give the client a free URL for the site.

The program will be written in Java, with most of the website using HTML, CSS, JS and JSP.

Java will be used for processing, and saving choices to Google's Datastore (cloud database), by importing the Datastore classes within the Google's Appengine plug-in for Eclipse. JSP will interact with the Java to get choices server side.

The web page will be created in HTML 5, using CSS for styling. JavaScript will be used for client side processing

Creating the program with Java and JSP would also make the Eclipse EE IDE particularly useful, due to its support for Java and JSP file types.

As a webpage in HTML 5, system requirements are minimal, with most devices with an internet connection being able to access it, including Smartphone's and tablets.

As only an Internet connection is required, the client side system requirements for Internet Explorer 7 will be used, as a staple core windows program for using the website. However the program itself will be optimised for Google Chrome due to its speed compared to IE7.

On a PC using IE7

Requirements
233MHz processor
Windows XP Service pack 2
Mouse or compatible pointing device
RAM (for the browser alone)
64MB for 32-bit windows XP/ Server 2003
128 MB for 64-bit windows XP/ Server 2003

The website will also work on OS X: 10.6, Android 4.0 or iOS 6.0 or later versions, when running Google Chrome.

## Nature of the Solution

### Design specification

2.

- a. The layout should be minimalistic and easy to use
- b. There should be a login screen first then a main menu with the three main options presented
- c. When creating characters the screen should bring up further choices
- d. Left to right, top down familiar design
- e. Use of complementary colours
- f. links to the main pages should be available on any part of the website in a bar at the top
- g. Centralised design
- h. Use of Pastel purple

These are identical to the design requirements layed out in the requirements specification. I will refer to them as I create my designs.

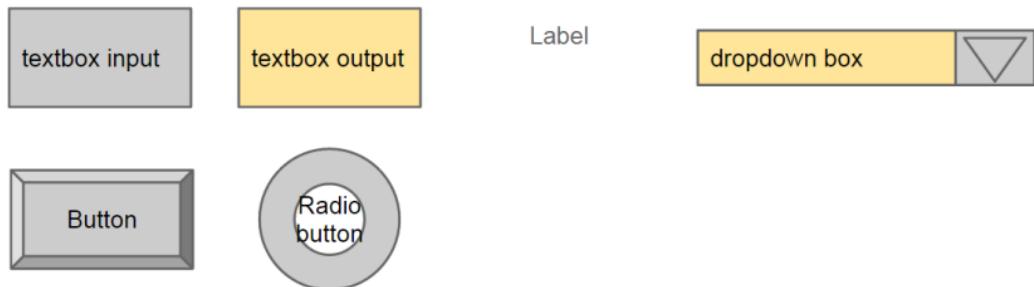
### Justification

- a. The client wanted the system to be as simple as possible for any new players wanting to make a character.
- b. The login screen is to allow saved characters to be accessed, then the three options are presented to give the user the freedom to immediately view characters rather than create a character should they want to.
- c. This will remove any excess unwanted information being given, which there will be a lot of.
- d. This is the standard for modern website design to make it consistent with layouts the user will be used to.
- e. Should make the website easy to look at and not an unpleasant experience.
- f. allows ease of access to the sites pages.
- g. keeps the information presented in the same general area, so the user will consistently know where to look.
- h. As stated in the analysis of the interview, the user specifically requested this colour to be a motif throughout the website, simply as his preference.

## Interface design

I began storyboarding the various pages of the website, following the requirements specification. To present the layout to Mr. Murray.

# key:



I first designed the login page and attempted to be as minimalistic as possible in my approach.

**Login**

Email Address	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Submit"/>	

[Password Recovery](#)

A link to the password recovery page, should any information be forgotten

A password recovery page is then necessary to avoid issues with forgotten information.

**Password recovery**

Email Address

After the E-mail address is entered, if that E-mail is attached to one of the users an E-mail with their password attached

## Character creation

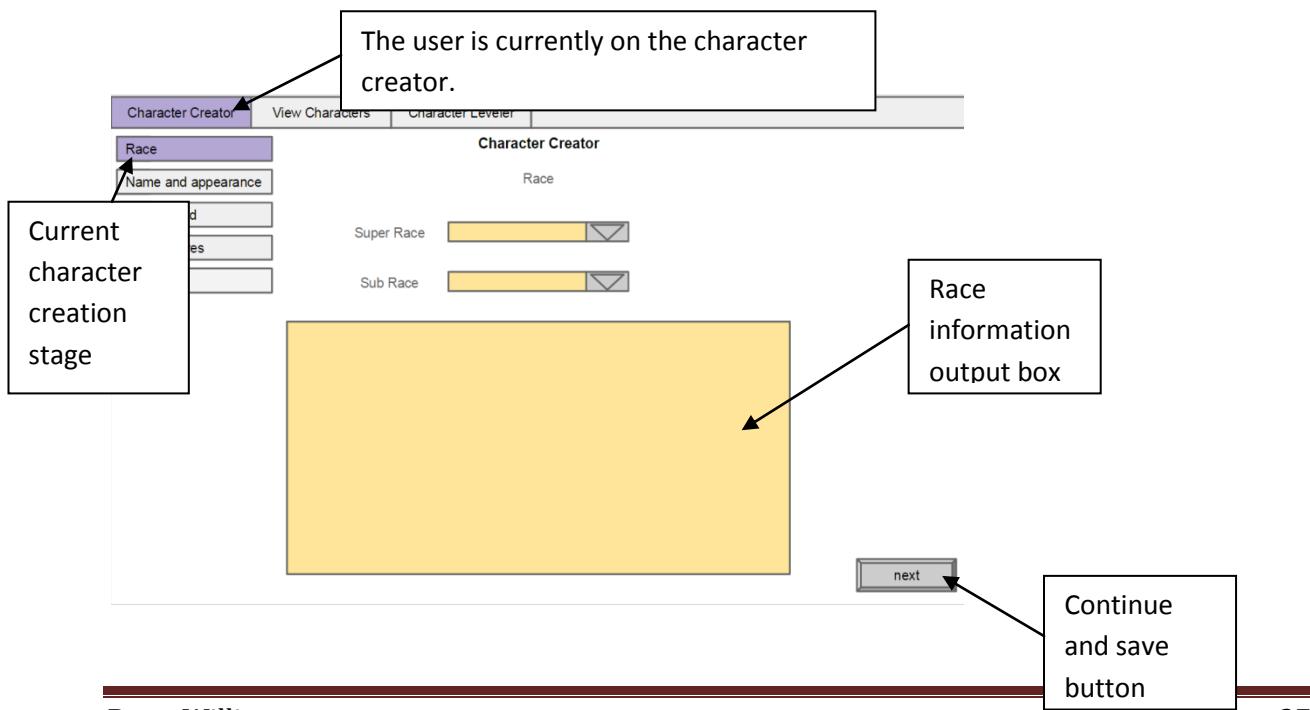
### Race choice

The first page that the users will come to is the character creator screen. The three main pages of the site are displayed at the top of every page, The page we are on is indicated in the designs by pale purple, however this will not be present in the final product, as the W3C standard for colour blindness means meaning shouldn't be conveyed by colour alone.

For the character creation page there is another set of tabs that show what stage of character creation the user is on and will allow them to jump to sections and append already completed sections.

The first character creation page will allow the user to select the race of their character and the sub-race should one be available, with dropdown boxes. Then when the user has chosen their race, the output box below the dropdown boxes will display important information about the racial features and an image of the race they have chosen.

At the bottom of each character creation page is a button that will validate choices, then save and take the user to the next stage.



## Name and physical characteristics choice

The next page allows the user to input all the physical characteristics of their character using textboxes.

The random height and weight button will use the random tables in the book for each race, so will use the information from the race page.

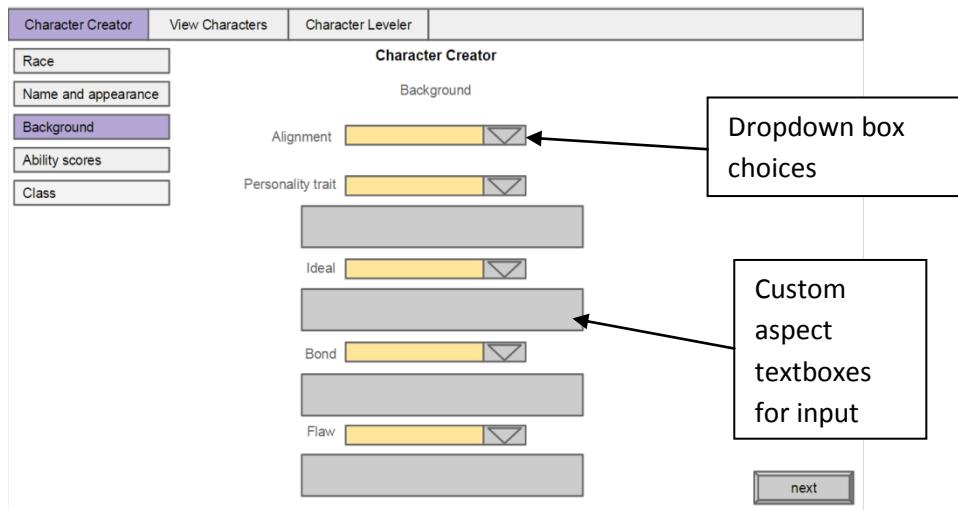
The screenshot shows the 'Character Creator' interface. On the left, there is a sidebar with buttons for Race, Name and appearance, Background, Ability scores, and Class. The 'Name and appearance' button is highlighted. The main area is titled 'Character Creator' and has sections for 'Name and appearance' and 'Characters name'. It includes input fields for 'Your actual name', 'Age', 'Eyes', 'Skin', 'Hair', 'Height', and 'Weight'. There are also buttons labeled 'Random height?' and 'Random Weight?'. A callout box labeled 'Input textboxes' points to the 'Your actual name' field. Another callout box labeled 'Random height and weight by race' points to the 'Random height?' and 'Random Weight?' buttons. A 'next' button is located at the bottom right.

## Background choice

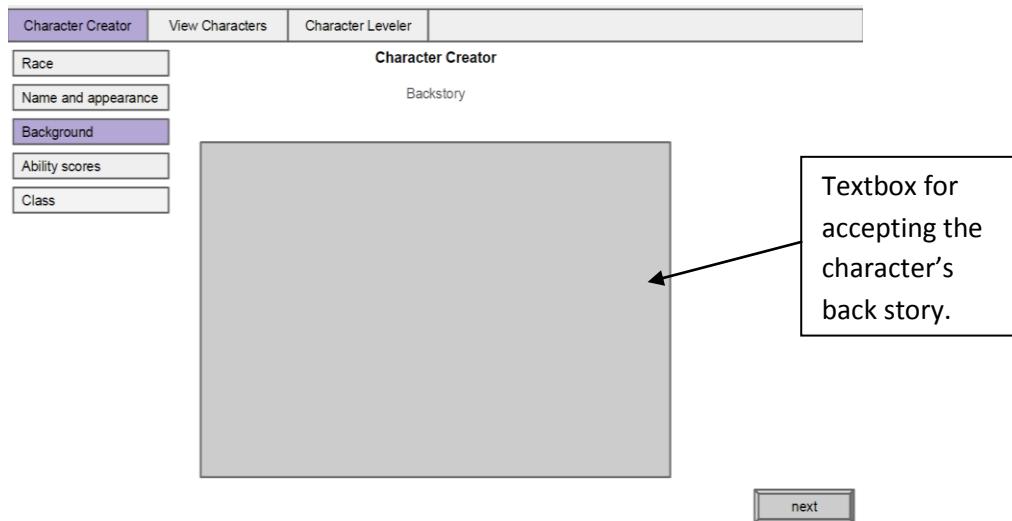
The next page allows the user to select their background via dropdown boxes from the backgrounds that are already available, then the benefits of the backgrounds are displayed below in the output boxes.

The screenshot shows the 'Character Creator' interface. On the left, there is a sidebar with buttons for Race, Name and appearance, Background, Ability scores, and Class. The 'Background' button is highlighted. The main area is titled 'Character Creator' and has sections for 'Background' and 'Type'. It includes dropdown menus for 'Background' and 'Type', and input fields for 'Skill proficiencies', 'Tool proficiencies', 'Languages', 'Equipment', and 'Feature'. A callout box labeled 'Background choice dropdown boxes' points to the 'Background' and 'Type' dropdowns. Another callout box labeled 'Output boxes for information about the benefits of the background' points to the 'Skill proficiencies', 'Tool proficiencies', 'Languages', 'Equipment', and 'Feature' fields. A 'next' button is located at the bottom right.

The next part of the background selection involves more options from dropdown boxes. However, choices such as personality traits don't have to be from the premade list. To deal with this, textboxes are present to allow custom aspects to be input by the user.



The final part of background selection is simply imputing text that describes the history of the created character. This is all user choice, so a large textbox is provided for all the information.



## Ability score choice

There are three methods of determining the ability scores of a player character:

- Rolling 4, 6 sided dice and ignoring the lowest result
- Using a standard set of numbers
- Being given a number of points with which the scores can be bought

Then when the users have their numbers, they can choose which ability scores get those values.

As such the first page allows the user to choose the method of determining ability scores

The screenshot shows the 'Character Creator' interface. At the top, there are tabs for 'Character Creator', 'View Characters', and 'Character Leveler'. Below these are dropdown menus for 'Race', 'Name and appearance', 'Background', 'Ability scores' (which is highlighted in purple), and 'Class'. The main area is titled 'Ability scores' and contains three buttons: 'Random?' (with a description: 'Use numbers as though you had rolled dice and used those values'), 'Preset?' (with a description: 'use the standard number'), and 'Point buy?' (with a description: 'use custom numbers by spending points'). A large callout box on the left points to the 'Ability scores' tab and the 'Random?' button, stating: 'Dropdown boxes allowing the user to choose their ability scores.' Another callout box on the right points to the 'Preset?' button, stating: 'The buttons take the user to the relevant page for generating their ability scores.'

The random method will generate 4 integers between 1 and 6 (as though four dice had been rolled). The lowest number is ignored. This will create a number between 3 and 18 with a slight bias for higher numbers. This will be done 6 times to create a number for each ability score. The user can then use the dropdown boxes to choose which ability scores get which numbers.

The output textbox on the right will display the 6 numbers available to the user. These numbers will also be present in the dropdown boxes, but as the user uses the numbers, they will be removed from the available numbers in other dropdown boxes.

The screenshot shows the 'Character Creator' interface with the 'Ability scores' tab selected. On the left, there are dropdown menus for 'Race', 'Name and appearance', 'Background', 'Ability scores' (highlighted in purple), and 'Class'. The main area is titled 'Random' and shows six dropdown boxes for the ability scores: Strength, Dexterity, Constitution, Intelligence, Wisdom, and Charisma. Each dropdown box has a yellow bar indicating the current value. To the right, a large callout box points to a text area labeled 'Your numbers are:' containing a list of numbers (dots) followed by a vertical ellipsis. Another callout box points to the 'next' button at the bottom right, stating: 'Textbox displaying all the numbers available to put in the ability score boxes.'

The second method of generating ability scores involves being given the standard set of numbers 15, 14, 13, 12, 10, 8. The user can then input these numbers into whatever ability scores they want using the dropdown boxes.

The screenshot shows the 'Character Creator' interface with a navigation bar at the top: 'Character Creator', 'View Characters', and 'Character Leveler'. On the left, there are dropdown boxes for 'Race', 'Name and appearance', 'Background', 'Ability scores' (which is selected), and 'Class'. In the center, under the heading 'Character Creator' and 'Preset', there are six dropdown boxes labeled 'Strength', 'Dexterity', 'Constitution', 'Intelligence', 'Wisdom', and 'Charisma'. To the right, a yellow box contains the text 'Your numbers are:' followed by a list of numbers: 15, 14, 13, 12, 10, 8. A callout box points from the 'Ability scores' dropdown to the Intelligence dropdown. Another callout box points from the list of numbers to the list itself. A 'next' button is at the bottom right.

Dropdown boxes allowing the user to choose their ability scores.

Textbox showing the standard array of ability scores

next

The final method of generating ability scores involves the user being allotted 27 points. They can then spend these points in order to buy numbers to use as ability scores. (Where higher numbers cost more points) The first dropdown box that the user selects an ability score from will decrement the remaining points by the cost of that score. The dropdown boxes will only show the ability scores a user can afford with their remaining points. Because the points for buying scores are dynamic, it will need to be updated as points are spent. A table showing how many points an ability score will cost is present to make the process easier to understand and keep a track of.

The screenshot shows the 'Character Creator' interface with a navigation bar at the top: 'Character Creator', 'View Characters', and 'Character Leveler'. On the left, there are dropdown boxes for 'Race', 'Name and appearance', 'Background', 'Ability scores' (selected), and 'Class'. In the center, under the heading 'Character Creator' and 'Point bought', there are six dropdown boxes labeled 'Strength', 'Dexterity', 'Constitution', 'Intelligence', 'Wisdom', and 'Charisma'. To the right, a table shows the cost of each ability score. A yellow box labeled 'Points remaining: 27' is shown above the table. Callout boxes point from the 'Ability scores' dropdown to the Intelligence dropdown and from the 'Points remaining' box to the table. A 'next' button is at the bottom right.

Dropdown boxes allowing the user to choose their ability scores.

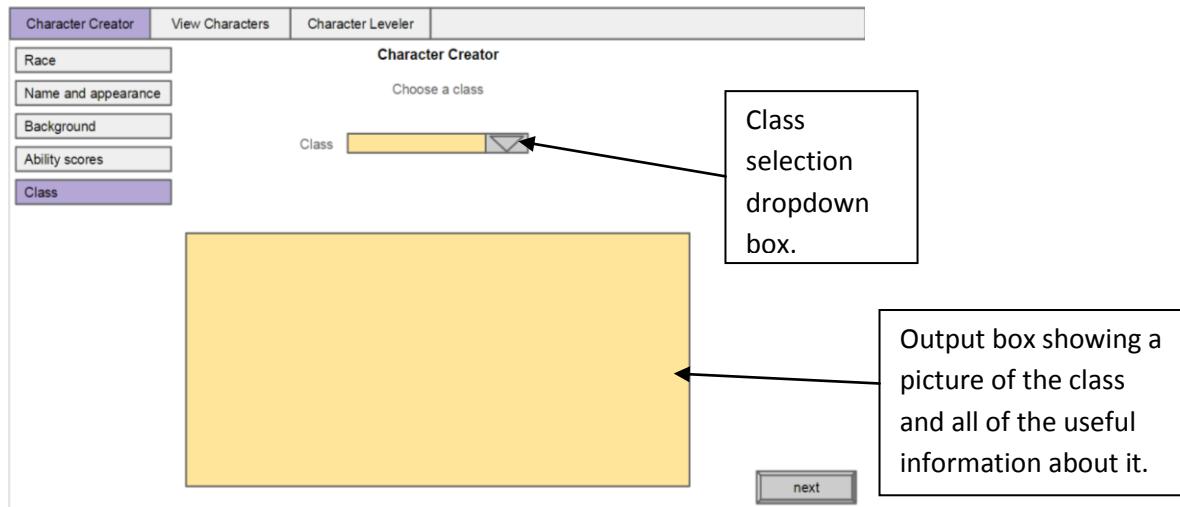
Textbox showing the users remaining unspent points, starting at 27.

Table showing ability scores and their point cost

next

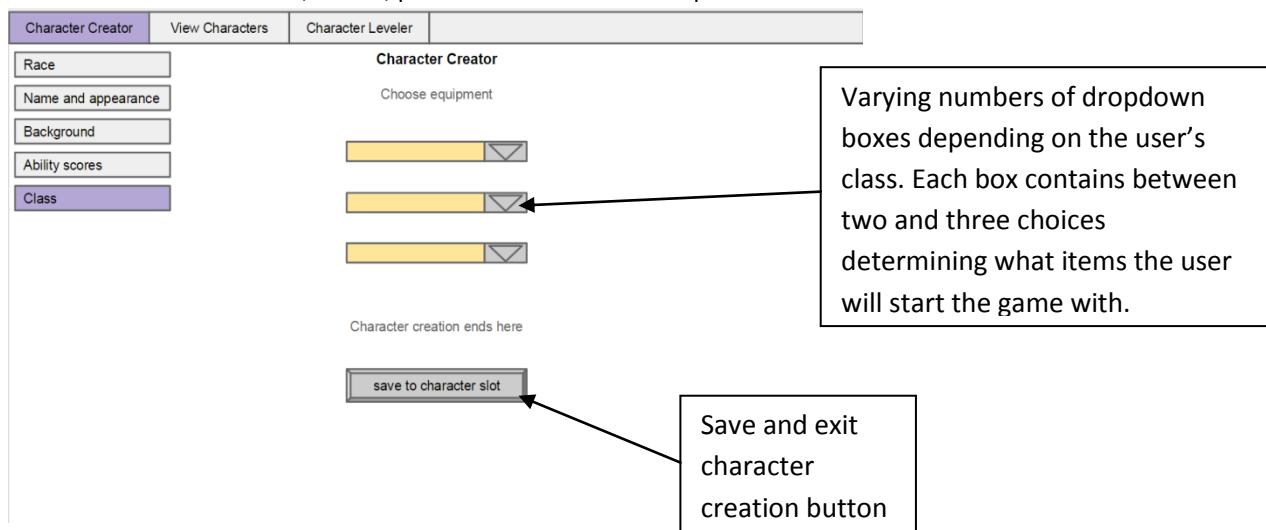
## Class choice

The final section of character creation allows the user to select what class they will be playing. This is done similarly to race, with a dropdown box for choosing the class and an output box for showing a picture of the class as well as all of the important information about it, such as class features.



The final stage of character creation involves choosing the starting equipment for the class. Depending on what class the user chose they can select different equipment to start with. But this also changes the number of dropdown boxes needed as some classes have fewer options than others.

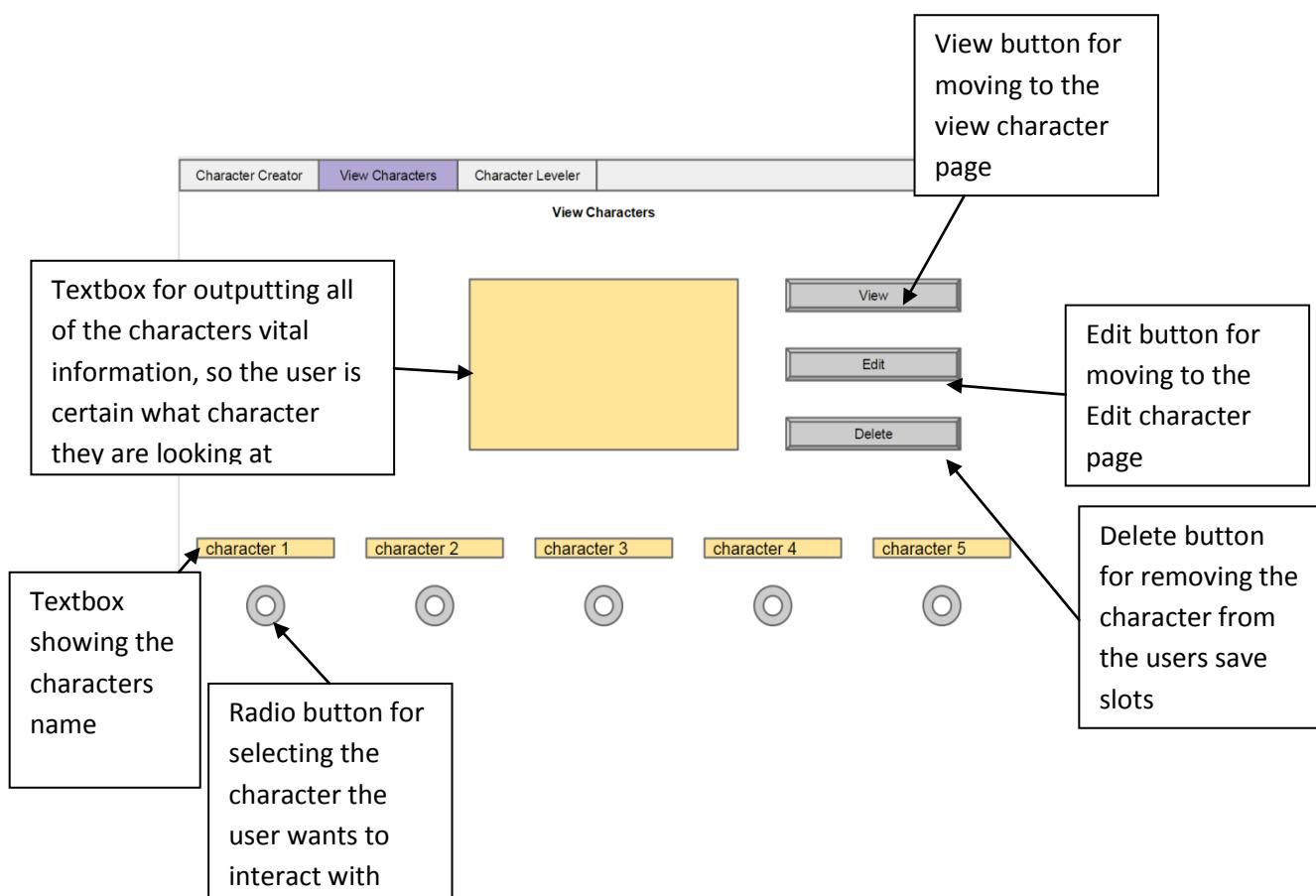
As the final stage of the character creation process this page has no button, instead a button that will leave the character creation page and save the character that has been created to a character slot attached to the users account so it can be viewed, edited, printed or deleted in other parts of the site.



## Character viewer

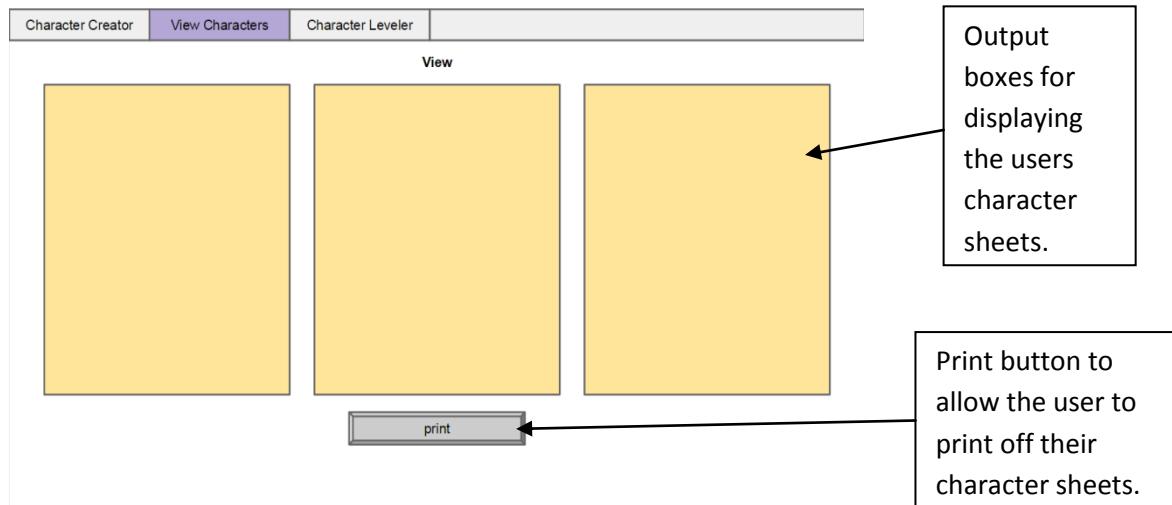
The user is moved to this page at the end of character creation, or by clicking on the tab at the top of the screen, which is then highlighted to indicate that the user is on the page.

Each user is allocated 5 character slots to their account. On this page they are displayed with their name and an accompanying radio button. Selecting a characters radio button will bring all of that characters information to the textbox at the centre of the screen, so the user can make sure they have selected the correct character. Then there are three buttons, one for viewing a character and enabling the user to print of their character sheet , another for editing a created character, and finally one for deleting the character in that slot.

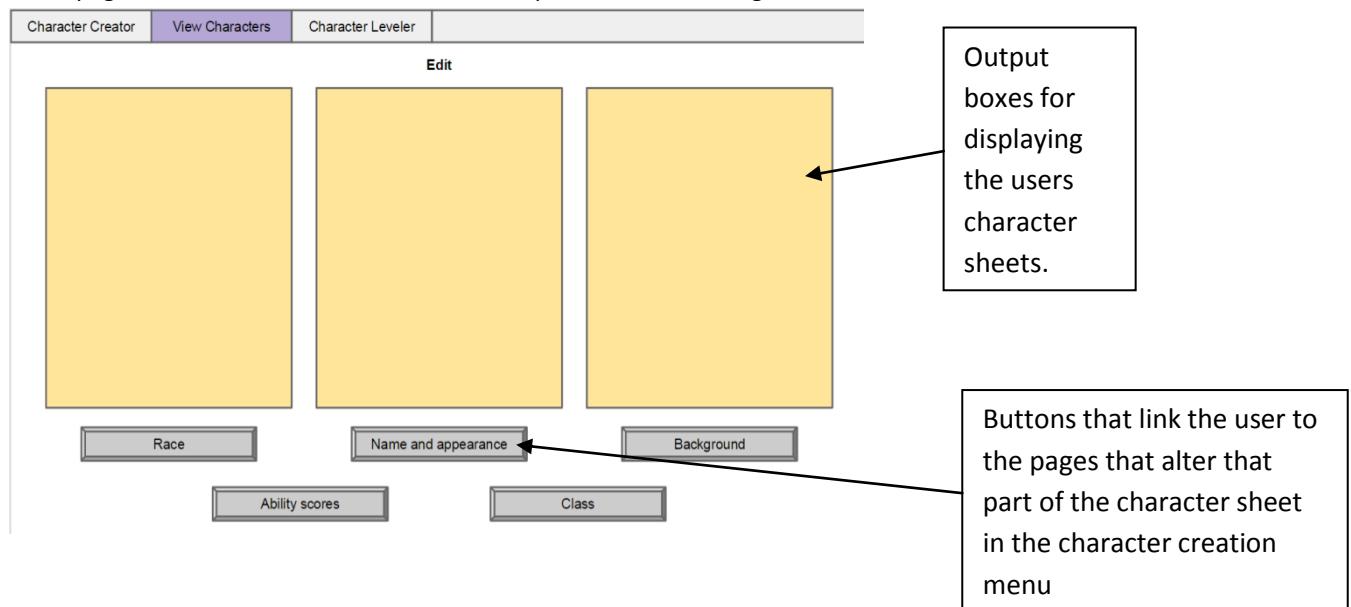


The view character page brings up the Character sheets for that character, as they would be used in a game. This allows the user to make sure that all of the information is correct.

A button then allows the user to print off their three character sheets.

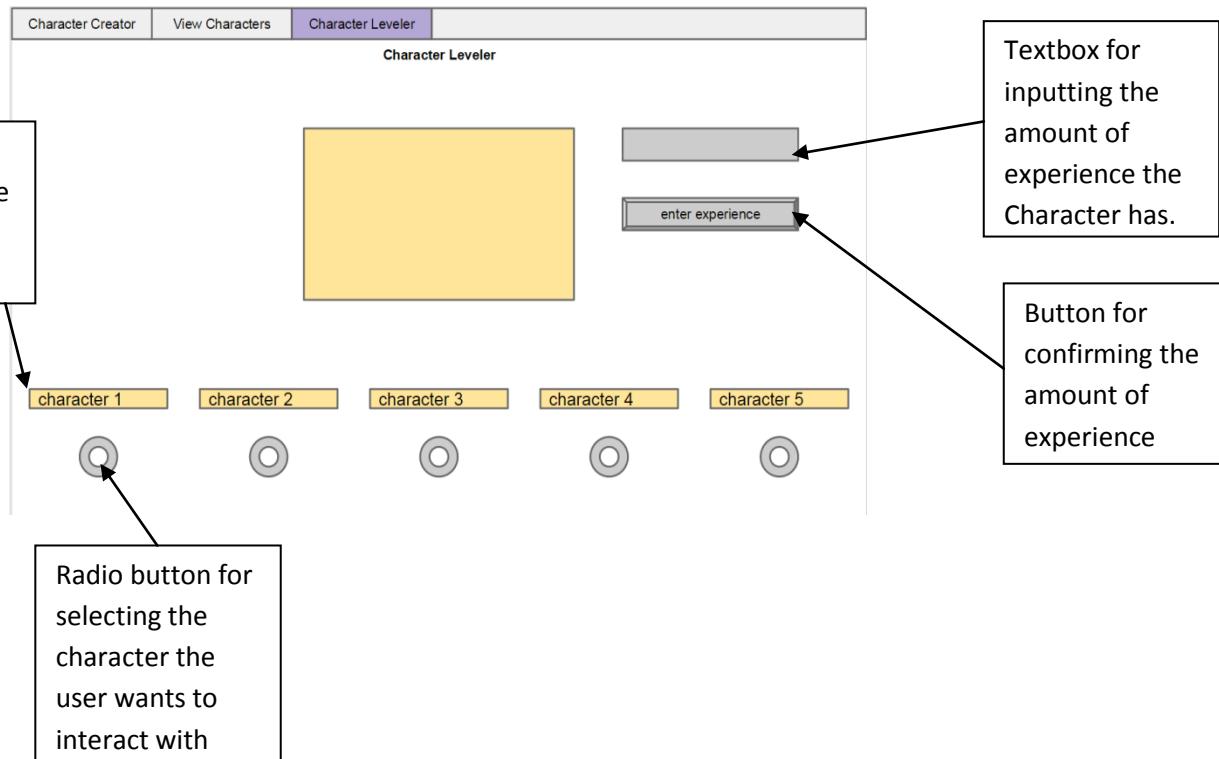


The edit page allows the user to see their character sheets as well as click buttons that will take them straight to the pages that allow the relevant data to be input in order to change that information.



## Character Leveller

The character leveller is the final tab at the top of the page. And can be accessed by clicking on it. It is similar in layout to the character viewer but allows the characters experience points to be input. This determines the characters level which alters many aspects of the character sheet, such as proficiency bonus and class features.



That is the first draught of how each webpage will look. I then sent this plan off to Mr. Murray for a response

Attachments: [interfaceDesign.docx \(4 MB\)](#) [Open as Web Page]

16 January 2015 13:14

Hello Liam,

I have finished creating the initial storyboard for the webpages as attached. I am keen to hear your thoughts and any changes you feel are needed

Kind regards,  
Bram

Hello Bram,

I am writing to give feedback on your preliminary designs. I found that Radio buttons as I believe they are called were by far the easiest way to make decisions online, if you agree I would like to focus on a more radio button based design.

Also on a separate note, we have realised that spells may be more easily done with the book due to the vast number of them, also character levelling can be left till last if we have time as its not as important as the initial creation.

Regards Liam.

Dear Liam,

I can replace a large number of the inputs with radio buttons very easily. I will be in touch soon with more developments

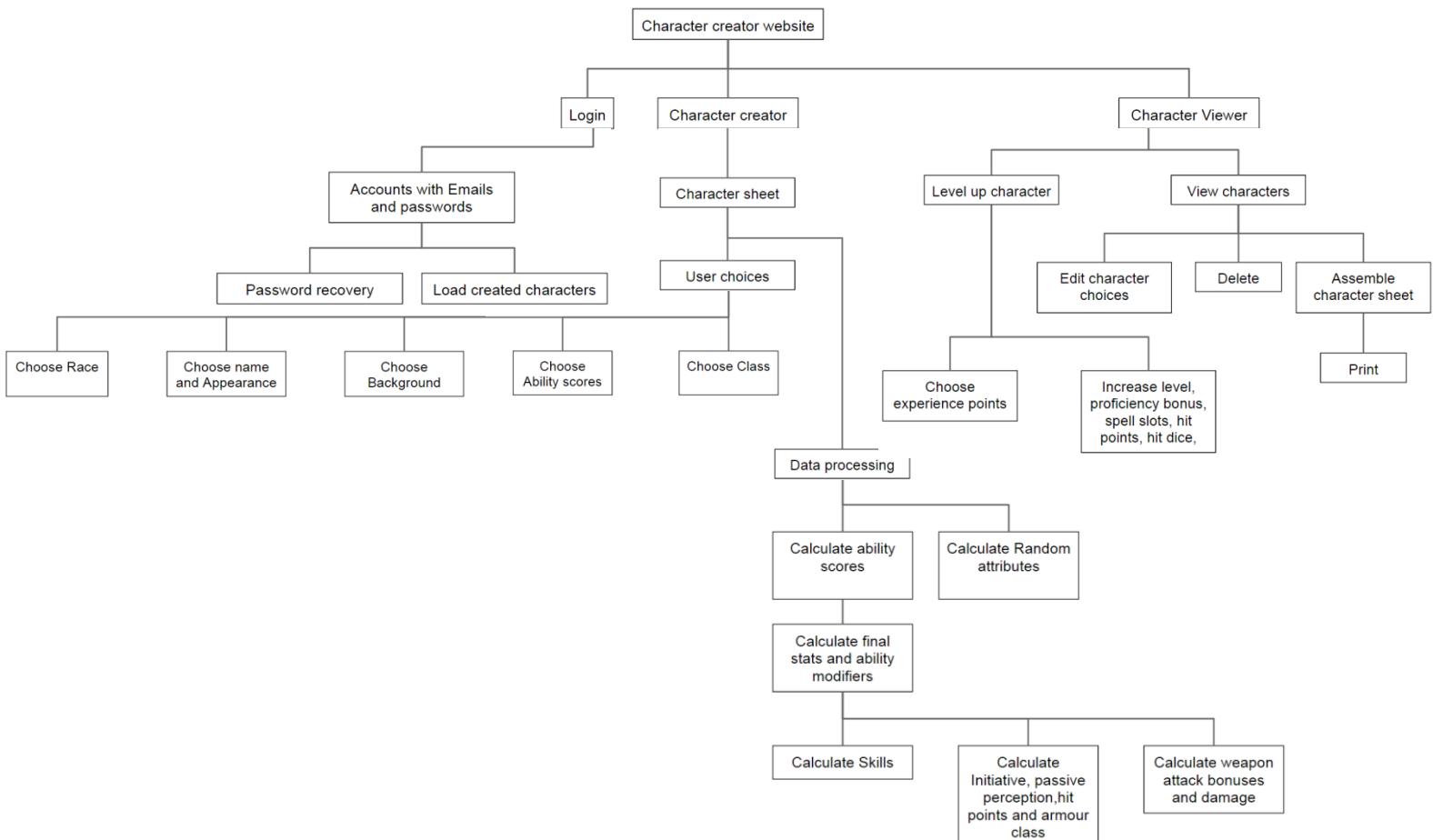
Kind regards,

Bram

With no Character leveller, the third tab of the navigation bar will need to be removed, as well as the character leveller pages.

# Algorithms

## Modular design



## Data structures

This system is going to require a database to store a lot of the information. Information that will need to be stored includes:

- Login information:
  - E-mail addresses
  - Passwords
  - Character slots
- Character information:
  - Races:
    - Subraces
    - Speed
    - Languages
    - Ability score increases
    - Proficiencies
    - Features
    - Age (bounds)
    - Height (bounds)
    - Weight (bounds)
  - Backgrounds:
    - Types
    - Proficiencies
    - Languages
    - Equipment
    - Personality traits
    - Ideals
    - Bonds
    - Flaws
    - Features
    - Alignments
  - Ability scores
    - Point buy cost table
    - Standard array
    - Convert to ability modifier  $((A-10)/2)$
    - Which proficiencies they affect
  - Class
    - Hit dice
    - Hit points at first level
    - Proficiencies
    - Equipment
    - Features
    - Spellcasting
- Character viewer:
  - Character sheet
- Character leveller:
  - Character information:
    - Level up experience table

- Proficiency bonus
- Spellcrafting
- Hitpoints
- Hidice

## Algorithms and pseudo code

There are multiple mathematical processes that will need to be repeated across the program.

### Created character class

A java class will be used to create objects of characters. Each object can hold the variables of all values these characters can have, and methods can be used to set the values of these variables depending on user choice. All the variables that the player will need will be layed out here. Initially it appeared that the variables should be storred in an array, however because of the nature of arrays, the values held by the arrays would be identical in every instance of the class, which we don't want. As a result no arrays are used for saving choices. The class:

```
class character{//class object to hold values when instanced

character{
}//constructor for objects of this class

//all the variables the class holds will be declared

String playerName//the name of the character
String name
String race//each race will need a setter method for other variables
String background//will need setters
String Class//will need setters
Int level = 1//at level 1

//ability scores, will need to be chosen, then modifiers and saves can be determined
Int Strength, StrengthModifier//modifiers are calculated from the final scores
Int Dexterity, DexterityModifier
Int Constitution, ConstitutionModifier
Int Intelligence, IntelligenceModifier
Int Wisdom, WisdomModifier
Int Charisma, CharismaModifier

//skills will need to be determined by ability modifiers and whether or not the character is proficient
//value holders, ability saves
Int StrengthSave//class determines what saves the player is proficient with
Int DexteritySave
Int ConstitutionSave
Int IntelligenceSave
Int WisdomSave
Int CharismaSave
//value holders, skills
```

```
Int athletics//calculated from the relevant ability modifier and adding 2 if the Boolean for the skill is true
Int acrobatics
Int sleightOfHand
Int stealth
Int arcane
Int history
Int investigation
Int nature
Int religion
Int animalHandling
Int insight
Int medicine
Int perception
Int Survival
Int deception
Int intimidation
Int performance
Int persuasion
//we will also need to save if the player is proficient in these skills, to calculate them properly and output to
the final character sheet
Boolean StrengthSave
Boolean DexteritySave
Boolean ConstitutionSave
Boolean IntelligenceSave
Boolean WisdomSave
Boolean CharismaSave
// skills
Boolean athletics
Boolean acrobatics
Boolean sleightOfHand
Boolean stealth
Boolean arcane
Boolean history
Boolean investigation
Boolean nature
Boolean religion
Boolean animalHandling
Boolean insight
Boolean medicine
Boolean perception
Boolean Survival
Boolean deception
Boolean intimidation
Boolean performance
Boolean persuasion
```

String eyes//these strings are for holding the characters appearance

String skin

String hair

```
String size
String height//string to allow for feet and inches notation
String weight//string to allow stone and pounds notation
Int age
String alignment
String personalityTraits//personality traits ideals bonds and flaws are chosen from a table for each background
String ideals
String bonds
String flaws
String backstory

String equipment
Int money//money and equipment are determined by background and class

String otherProficiencies//this can be altered by race, background and class
Int speed//this depends on race
String vision//this depends on race
String languages//this can be altered by race, background and class
String features//this can be altered by race, background and class
String feats
Int level
String hitDice
Int hitPoints//this and hit dice depends on the class and constitution modifier
Int ArmourClass//this can be used in a switch case block to determine the characters armour class
Int initiative//this is equal to the dexterity modifier
Int passivePerception//this is the perception skill value plus 10

String SpellAbility//all spell variables depend on class
Int SpellSave
Int SpellModifier
String Spells
//the object will also hold methods that when run set the object variables appropriately
}
```

## Race setter methods

In order for the user to make choices, setter methods will be used to alter the values of the variables of their instance of the class. The first choice the user will make is selecting their race. These are generally similar to the races of lord of the rings and most fantasy genres. Including humans, dwarves and elves. Each race has different features that alters the player. All the effects of every choice can be found in the RPGs players handbook.

Race choices:

1. High elf
2. Wood elf

3. Dark elf
4. Hill dwarf
5. Mountain dwarf
6. Lightfoot halfling
7. Stout halfling
8. Human / variant (similar but with different abilities)
9. Dragonborn
10. Forest gnome
11. Rock Gnome
12. Half-elf
13. Half-orc
14. Tiefling

The choice of race alters many things. All races alter:

- the race String
- the ability score integers (each race alters the ability scores differently, normally increasing two values)
- the speed integer
- the vision String
- the languages String
- the size String

Each race also has specific values for calculating the players weight and height. They have a base Height, a base weight, a height modifier and a weight modifier.

Also the choice or race can effect:

- the features String
- the other Proficiencies String
- skill proficiency booleans

For example the wood elf rules as shown by the book are:

## ELF TRAITS

Your elf character has a variety of natural abilities, the result of thousands of years of elven refinement.

**Ability Score Increase.** Your Dexterity score increases by 2.

**Age.** Although elves reach physical maturity at about the same age as humans, the elven understanding of adulthood goes beyond physical growth to encompass worldly experience. An elf typically claims adulthood and an adult name around the age of 100 and can live to be 750 years old.

**Alignment.** Elves love freedom, variety, and self-expression, so they lean strongly toward the gentler aspects of chaos. They value and protect others' freedom as well as their own, and they are more often good than not.

**Size.** Elves range from under 5 to over 6 feet tall and have slender builds. Your size is Medium.

**Speed.** Your base walking speed is 30 feet.

**Darkvision.** Accustomed to twilit forests and the night sky, you have superior vision in dark and dim conditions. You can see in dim light within 60 feet of you as if it were bright light, and in darkness as if it were dim light. You can't discern color in darkness, only shades of gray.

**Keen Senses.** You have proficiency in the Perception skill.

**Fey Ancestry.** You have advantage on saving throws against being charmed, and magic can't put you to sleep.

**Trance.** Elves don't need to sleep. Instead, they meditate deeply, remaining semiconscious, for 4 hours a day. (The Common word for such meditation is "trance.") While meditating, you can dream after a fashion; such dreams are actually mental exercises that have become reflexive through years of practice. After resting in this way, you gain the same benefit that a human does from 8 hours of sleep.

**Languages.** You can speak, read, and write Common and Elvish. Elvish is fluid, with subtle intonations and intricate grammar. Elven literature is rich and varied, and their songs and poems are famous among other races. Many bards learn their language so they can add Elvish ballads to their repertoires.

## WOOD ELF

As a wood elf, you have keen senses and intuition, and your fleet feet carry you quickly and stealthily through your native forests. This category includes the wild elves (grugach) of Greyhawk and the Kagonesti of Dragonlance, as well as the races called wood elves in Greyhawk and the Forgotten Realms. In Faerûn, wood elves (also called wild elves, green elves, or forest elves) are reclusive and distrusting of non-elves.

Wood elves' skin tends to be copperish in hue, sometimes with traces of green. Their hair tends toward browns and blacks, but it is occasionally blond or copper-colored. Their eyes are green, brown, or hazel.

**Ability Score Increase.** Your Wisdom score increases by 1.

**Elf Weapon Training.** You have proficiency with the longsword, shortsword, shortbow, and longbow.

**Fleet of Foot.** Your base walking speed increases to 35 feet.

**Mask of the Wild.** You can attempt to hide even when you are only lightly obscured by foliage, heavy rain, falling snow, mist, and other natural phenomena.

and thusly the wood elf setter would be:

```
woodElf(){  
this race = "Wood Elf"  
this dexterity = this dexterity + 2  
this wisdom = this wisdom + 2  
this vision = "Darkvision"  
this languages = "common, elf"  
this features = "trance, fey ancestry"  
this otherProficiencies = "longbow, shortbow, longsword, shortsword"  
this features = mask of the wild  
this speed = 35  
this baseHeight = 54//these values are found elsewhere in the book  
this heightModifier = random(2-20)random number from 2 to 20  
this baseWeight = 100
```

```
this weightModifier = random(1-4)random number from 1 to 4  
}
```

## Background Setter methods

The second choice the user will make is their characters background. This is what the character did before they became an adventurer. These have various effect on the skills the player has, and their personality.

Background choices:

1. Acolyte
2. Charlatan
3. Criminal
4. Spy
5. Entertainer
6. Gladiator
7. Folk Hero
8. Guild Artisan
9. Guild Merchant
10. Hermit
11. Noble
12. Knight
13. Outlander
14. Sage
15. Sailor
16. Pirate
17. Soldier
18. Urchin

The background choice alters multiple base variables:

- The background String
- Skill proficiency booleans
- The money int
- The equipment String
- The features String

Depending on the background it can also alter:

- The languages String
- The otherProficiencies String

The Outlander background choice in the book looks like:

## **OUTLANDER**

You grew up in the wilds, far from civilization and the comforts of town and technology. You've witnessed the migration of herds larger than forests, survived weather more extreme than any city-dweller could comprehend, and enjoyed the solitude of being the only thinking creature for miles in any direction. The wilds are in your blood, whether you were a nomad, an explorer, a recluse, a hunter-gatherer, or even a marauder. Even in places where you don't know the specific features of the terrain, you know the ways of the wild.

**Skill Proficiencies:** Athletics, Survival

**Tool Proficiencies:** One type of musical instrument

**Languages:** One of your choice

**Equipment:** A staff, a hunting trap, a trophy from an animal you killed, a set of traveler's clothes, and a belt pouch containing 10 gp

### **FEATURE: WANDERER**

You have an excellent memory for maps and geography, and you can always recall the general layout of terrain, settlements, and other features around you. In addition, you can find food and fresh water for yourself and up to five other people each day, provided that the land offers berries, small game, water, and so forth.

### **SUGGESTED CHARACTERISTICS**

Often considered rude and uncouth among civilized folk, outlanders have little respect for the niceties of life in the cities. The ties of tribe, clan, family, and the natural world of which they are a part are the most important bonds to most outlanders.

There is also a page showing the options for personality traits, ideals bonds and flaws but this will be set in another algorithm.

and the outlander background setter method

```
outlander(instrument, language){  
    this background = "outlander"  
    this athletics = true  
    this survival = true  
    this language = this languages + language  
    this otherProficiencies = this otherProficiencies + instrument  
    this money = 10  
    this equipment = this equipment + "A staff, a hunting trap, a trophy from an animal you killed, a set of travellers clothes"  
    this features = this features + "wanderer"  
}
```

## Class Setter methods

The final major choice the user can make is their character class. This is the role they play in a group e.g. fighter, wizard, ranger etc

Class choices:

1. Barbarian
2. Bard
3. Cleric
4. Druid
5. Fighter
6. Monk
7. Paladin
8. Ranger
9. Rogue
10. Sorceror
11. Warlock
12. Wizard

The class choice alters multiple base variables:

- The class String
- The hitDice String
- The hitPoint Int
- The otherProficiencies String
- The ability save booleans
- Skill proficiency booleans
- The equipment String
- The features String
- The Armour class int

It can also alter:

- The SpellAbility String
- the SpellSave Int
- The SpellModifier Int
- The Spells String

The fighter class page looks like this:

## CLASS FEATURES

As a fighter, you gain the following class features.

### HIT POINTS

**Hit Dice:** 1d10 per fighter level

**Hit Points at 1st Level:** 10 + your Constitution modifier

**Hit Points at Higher Levels:** 1d10 (or 6) + your Constitution modifier per fighter level after 1st

### PROFICIENCIES

**Armor:** All armor, shields

**Weapons:** Simple weapons, martial weapons

**Tools:** None

**Saving Throws:** Strength, Constitution

**Skills:** Choose two skills from Acrobatics, Animal Handling, Athletics, History, Insight, Intimidation, Perception, and Survival

### EQUIPMENT

You start with the following equipment, in addition to the equipment granted by your background:

- (a) chain mail or (b) leather, longbow, and 20 arrows
- (a) a martial weapon and a shield or (b) two martial weapons
- (a) a light crossbow and 20 bolts or (b) two handaxes
- (a) a dungeoneer's pack or (b) an explorer's pack

### FIGHTING STYLE

You adopt a particular style of fighting as your specialty. Choose one of the following options. You can't take a Fighting Style option more than once, even if you later get to choose again.

#### ARCHERY

You gain a +2 bonus to attack rolls you make with ranged weapons.

#### DEFENSE

While you are wearing armor, you gain a +1 bonus to AC.

#### DUELING

When you are wielding a melee weapon in one hand and no other weapons, you gain a +2 bonus to damage rolls with that weapon.

#### GREAT WEAPON FIGHTING

When you roll a 1 or 2 on a damage die for an attack you make with a melee weapon that you are wielding with two hands, you can reroll the die and must use the new roll, even if the new roll is a 1 or a 2. The weapon must have the two-handed or versatile property for you to gain this benefit.

#### PROTECTION

When a creature you can see attacks a target other than you that is within 5 feet of you, you can use your reaction to impose disadvantage on the attack roll. You must be wielding a shield.

#### TWO-WEAPON FIGHTING

When you engage in two-weapon fighting, you can add your ability modifier to the damage of the second attack.

### SECOND WIND

You have a limited well of stamina that you can draw on to protect yourself from harm. On your turn, you can use a bonus action to regain hit points equal to  $1d10 + \text{your fighter level}$ .

Once you use this feature, you must finish a short or long rest before you can use it again.

For example the fighter class setter method

```
Fighter(skill1, skill2, item1, item2, item3, item4, item5, feature){  
    this class = "fighter"  
  
    this hitDice = level + "D10" //hit dice are measured in a number of dice, at level 1 for a fighter it is 1 ten sided dice  
  
    this hitPoints = 10 + ConstitutionModifier //constitution modifier is the modifier of the constitution ability score. This is calculated by another ability score  
    this otherProficiencies = this otherProficiencies + ", all armour, shields, simple weapons, martial weapons"  
    this StrengthSave = true  
    this constitutionSave = true  
  
    switch(skill1){//skill1 is an integer chosen by the user, representing the skill they want to be proficient in  
        case 1: this acrobatics = true  
        break  
        case 2: this animalHandling = true  
        break  
        case 3: this athletics = true  
    }  
}
```

```

break
case 4: this history = true
break
case 5: this insight = true
break
case 6: this intimidation = true
break
case 7: this perception = true
break
case 8: this survival = true
break
}
switch(skill2){//skill2 is an integer chosen by the user, representing the skill they want to be proficient in, from
the same list as skill1
case 1: this acrobatics = true
break
case 2: this animalHandling = true
break
case 3: this athletics = true
break
case 4: this history = true
break
case 5: this insight = true
break
case 6: this intimidation = true
break
case 7: this perception = true
break
case 8: this survival = true
break
}
//the user then gets multiple choices for what equipment they want
//first chainmail armour or leather armour and a bow, as the choice is binary, the variable can be a Boolean
if(item1 = true){
this equipment = this equipment + "chainmail"
this armour = 7//a variable representing chainmail armour, for processing armour class later on, the value is a
place holder
}else{
this equipment = this equipment + "leather armour, longbow, 20 arrows"
this armour = 2//another placeholder value
}
//the second choice is a martial weapon and shield or two martial weapons. So if only one martial weapon is
chosen, we know that they also need a shield. So:
if(item3 == ""){
this equipment = this equipment + item2 + ", shield"
}else{
this equipment = this equipment + item2 + ", " + item3
}
//then the user chooses a light crossbow or handaxes, a Boolean choice, we will treat the light crossbow as

```

```

true for item4
if(item4 == true){
this equipment = this equipment +"light crossbow and 20 bolts"
}else{
this equipment = this equipment + "two handaxes"
}
//the final item is a pack, another Boolean choice. We will treat dungeoneers pack as true
if(item5 == true){
this equipment = this equipment + "dungeoneers pack"
}else{
this equipment = this equipment + "explorers pack"
}
//finally the feature choice, this can be an integer interacting with a switch cas block. The only feature
affecting the character sheet is defence which will increase the users armour class in another algorithm. As
such it will alter a Boolean value
switch(feature){
case 1: this features = this features + "archery"
break
case 2: this features = this features + "defence"
this defence = true
break
case 3: this features = this features + "duelling"
break
case 4: this features = this features + "great weapon fighting"
break
case 5: this features = this features + "protection"
break
case 6: this features = this features + "two weapon fighting"
break
}
this features = this features + "second wind"//all fighters get second wind
}

```

Each ability score, alignment, appearance and backstory will also need to be selected by the user with their own methods. Setting these will be simple with a method that accepts a value and sets it to the appropriate local variable e.g.

```

setStrengthScore(scoreIn){
this StrengthScore = scoreIn
}

```

### **Random ability score**

when finding random ability scores, the 4d6 method is used, where four six sided dice are rolled and their sum is taken ignoring the lowest value:

```

Int Dice[] = {radom(1-6), radom(1-6), radom(1-6), radom(1-6)}
int y=7
Int randScore

```

```

For( x=0; x<Dice.length; x++){
    if Dice[x]<y
        y = Dice[x]
}

```

randScore = Dice[0] + Dice[1] + Dice[2] + Dice[3] -y // as y is the lowest value, this will ignore the lowest value as it cancels out.

randScore can now be used to determine an ability score.

### Ability modifier

Ability scores exist for some functions and to create a value called the Ability modifier, this affects all the players dice rolls and is an integral part of the game. The ability modifier is found by subtracting ten from the ability score and dividing the result by two, then rounding down the result. As shown by this table:

ABILITY SCORES AND MODIFIERS			
Score	Modifier	Score	Modifier
1	-5	16–17	+3
2–3	-4	18–19	+4
4–5	-3	20–21	+5
6–7	-2	22–23	+6
8–9	-1	24–25	+7
10–11	+0	26–27	+8
12–13	+1	28–29	+9
14–15	+2	30	+10

Int abilityScore

Int abilityModifier = (abilityScore-10)/2

However in the clients role playing game, modifiers are always rounded down. Which will work for positive values, however negative modifiers will need to be manually rounded down.

### Random Heights and weights

All the races have the same method for generating random heights and weights, however depending on the race they have different ranges. For example a human has a base height of 4' 8", then the result of two 10 sided dice is added in inches. This equates to a random number from 2 to 20 being added. Then the weight is the increase in height, multiplied by the sum of two four sided dice added to a base weight of 110 lbs. As a result when the race is chosen, the appropriate base weight, height and modifier will need to be selected //for a human the base and modifiers will have already been set to

BaseHeight =56// inches

BaseWeight = 110// pounds

heightModifier = random(2-16)//two eight sided dice, two to 16

weightModifier = random(2-12)//two 6 sided dice, two to 12

Height = BaseHeight + heightModifier

Weight = BaseWeight + (heightModifier \* weightModifier)

String HeightImperial = (Height /12) + " " + (Height % 12) + """/convert to an imperial unit string

String WeightImperial = (Weight /14) + " stone " + (Weight % 14) + " pounds"

The function remains the same but depending on the race, the base and modifier values change.

## Calculate skill modifiers

The 18 different skills are calculated by summing the ability modifier the skill relies on with the characters proficiency bonus, if they are proficient with that skill

each skill depends on a particular ability score, as shown by this table:

<b>Strength</b>	<b>Wisdom</b>
Athletics	Animal Handling
<b>Dexterity</b>	<b>Charisma</b>
Acrobatics	Insight
Sleight of Hand	Medicine
Stealth	Perception
<b>Intelligence</b>	Survival
Arcana	
History	
Investigation	
Nature	
Religion	

Prof = 2// proficiency bonus, +2 at level one

StrengthModifier//modifiers calculated from scores, usually from -1 to +4

dexterityModifier

constitutionModifier//doesn't determine any skills

intelligenceModifier

wisdomModifier

charismaModifier

//proficiency values, hold the value of the skill after calculation

athleticsV

acrobaticsV

sleightOfHandV

stealthV

arcanaV

historyV

investigationV

natureV

religionV

animalHandlingV

insightV

medicineV

perceptionV

survivalV

deceptionV

intimidationV

performanceV

persuasionV

//proficiency Booleans, all true or false, determine whether or not to add the proficiency bonus  
athletics//proficiency is controlled by either background race or class.

acrobatics  
sleightOfHand  
stealth  
arcana  
history  
investigation  
nature  
religion  
animalHandling  
insight  
medicine  
perception  
survival  
deception  
intimidation  
performance  
persuasion

//calculations, if the Boolean is true add the proficiency bonuses

// dependent on strength

If(athletics == true){athleticsV = StrengthModifier + prof}

// dependent on dexterity

If(acrobatics == true){ acrobaticsV = dexterityModifier + prof}

If(sleightOfHand == true){ sleightOfHandV = dexterityModifier + prof}

If(stealth == true){stealthV = dexterityModifier + prof}

//dependent ion intelligence

If(arcana == true){ arcanaV = intelligenceModifier + prof}

If(history == true){ historyV = intelligenceModifier + prof}

If(investigation == true){ investigationV = intelligenceModifier + prof}

If(nature == true){ natureV = intelligenceModifier + prof}

If(religion == true){ religionV = intelligenceModifier + prof}

//dependent on wisdom

If(animalHandling == true){ animalHandlingV = wisdomModifier + prof}

If(insight == true){ insightV = wisdomModifier + prof}

If(medicine == true){ medicineV = wisdomModifier + prof}

If(perception == true){ perceptionV = wisdomModifier + prof}

If(survival == true){ survivalV = wisdomModifier + prof}

//dependent on charisma

If(deception == true){ deceptionV = charismaModifier + prof}

If(intimidation == true){ intimidationV = charismaModifier + prof}

If(performance == true){ performanceV = charismaModifier + prof}

If(persuasion == true){ persuasionV = charismaModifier + prof}

Certain choices can give player characters expertise in a skill. This will allow the character to have their proficiency bonus doubled in certain skills. In order to double skills. A second array of Boolean values will be needed to say if the proficiency should be doubled for a skill

## Calculate Armour Class

A characters armour class depends on a few things. The players armour, whether or not they have a shield, and depending on their armour, their dexterity modifier. Also the defence feature for fighters can increase the players armour class. If a player is wearing light armour, their armour class is a base value plus their dexterity modifier. If they wear medium armour, it is a higher base value, but the bonus from dexterity is capped to 2. Finally heavy armour gives a flat armour class score. On top of these, having a shield gives the player a bonus of two to their armour class, and the fighter feature gives one.

As the player either has the shield or not, and has the fighter trait or not, it makes sense for them to be Booleans, that add the bonus at the end if true.

The armour is layed out thusly.

Armor	Cost	Armor Class (AC)	Strength	Stealth	Weight
<i>Light Armor</i>					
Padded	5 gp	11 + Dex modifier	—	Disadvantage	8 lb.
Leather	10 gp	11 + Dex modifier	—	—	10 lb.
Studded leather	45 gp	12 + Dex modifier	—	—	13 lb.
<i>Medium Armor</i>					
Hide	10 gp	12 + Dex modifier (max 2)	—	—	12 lb.
Chain shirt	50 gp	13 + Dex modifier (max 2)	—	—	20 lb.
Scale mail	50 gp	14 + Dex modifier (max 2)	—	Disadvantage	45 lb.
Breastplate	400 gp	14 + Dex modifier (max 2)	—	—	20 lb.
Half plate	750 gp	15 + Dex modifier (max 2)	—	Disadvantage	40 lb.
<i>Heavy Armor</i>					
Ring mail	30 gp	14	—	Disadvantage	40 lb.
Chain mail	75 gp	16	Str 13	Disadvantage	55 lb.
Splint	200 gp	17	Str 15	Disadvantage	60 lb.
Plate	1,500 gp	18	Str 15	Disadvantage	65 lb.
<i>Shield</i>					
Shield	10 gp	+2	—	—	6 lb.

armourClass

dexterityModifier

armour //after class choice this instance of armour should be set to whatever armour the player

shield //Boolean

defence//Boolean

switch(armour){

case 0://no armour

this armourClass = 10 + dexterityModifier

break

case 1://case 1, 2 and 3 will be light armours, with 1 being padded

this armourClass = 11 + dexterityModifier

break

case 2:// leather

this armourClass = 11 + dexterityModifier

break

case 3://studded

```

this armourClass = 12 + dexterityModifier
break
//medium armours
case 11://cases 11 to 15 will be medium armours, extra functionality for dexterity modifiers above 2 is
necessary for these
if(this dexterityModifier > 2){//if the modifier is greater than 2 only add 2
this armourClass = 12 + 2
}else{
this armourClass = 12 + dexterityModifier
}
break
case 12://hide armour
if(this dexterityModifier > 2){//if the modifier is greater than 2 only add 2
this armourClass = 13 + 2
}else{
this armourClass = 13 + dexterityModifier
}
break
case 13://scale armour
if(this dexterityModifier > 2){//if the modifier is greater than 2 only add 2
this armourClass = 14 + 2
}else{
this armourClass = 14 + dexterityModifier
}
break
}
case 14://breastplate
if(this dexterityModifier > 2){//if the modifier is greater than 2 only add 2
this armourClass = 14 + 2
}else{
this armourClass = 14 + dexterityModifier
}
break
}
case 15://half platte
if(this dexterityModifier > 2){//if the modifier is greater than 2 only add 2
this armourClass = 15 + 2
}else{
this armourClass = 15 + dexterityModifier
}
break
//heavy armours, no dexterity bonus
case 21://ringmail
this armourClass = 14
break
case 22://chainmail
this armourClass = 16
break
case 23://splint

```

```

this armourClass = 17
break
case 24://plate
this armourClass = 18
break
}
//finally the bonus for the shield and fighter feature need to be added
if(this shield == true){
this ac = this ac + 2
}
if(this defence == true){
this ac = this ac + 1
}

```

## Login Database

As The Google Appengine plugin for eclipse is going to be used, the google datastore cloud database will be used. This is useful as it allows the creation of tables with a number of columns that can vary from entity to entity, meaning the program can be updated without altering past saved data.

The datastore has tables known as ‘kinds’ two will be used. The first will hold the user’s username and password. The second will use the username from the first kind as a foreign key, and hold characters tied to users.

“Login” kind:

username	password

each row will be populated with created usernames and passwords

“character” kind

## login functions

A java helper class will be needed to help with certain parts of interacting with googles datastore. The first function will check if a username is free, to let another function know if an account can be created with that name. As such the function will return a Boolean, and need the name being searched for as a parameter.

```

helper{
Boolean checkNameFree (nameIn){
Datastore datastore = getDatastoreService
Filter filter = new filter("username", check equal, nameIn)
Query nameFree = new Query("login") setFilter(filter)
Entity entity = datastore nameFree
if(entity = null){

```

```

return true
}else{
return false
}
}
}

```

This creates a filter, finding any user with the nameIn parameter and applies it to the database. If any are found it will not return null and the function will return false, saying that name cant be used. However if it finds null with that username, true is returned as the name can be used

Now we know if the name is free, we need a function to add a username and password to the login page. This can be done in a jsp page.

On the page there should be a form called, username password1 and password2. These will allow the user to submit a username and confirmed password to be put into the datastore. With the following algorithm.

```

import helper class
if(request("username") != null & request("password1") != null & request("password2") != null){//on form submit
u = request("username")
P1 = request("username")
P2 = request("username")
if(helper.nameFree(u) == true){//if the name is free to use
if(p1 == p2){//if the two passwords are identical
Datastore datastore = getDataservice
Entity entity = new entity ("login")//create a new entity in the login kind
entity set property ("username", u)//create a column called username and put the input username in the row
entity set property ("password", p1)//create a password column, and put the input password in the row
datastore put user//put the entity in the datastore
}
}
}

```

this will create the entity for the user to login with. Logging in will involve getting the usernames from the datastore, if one with the input username exists, check if the password entered matches the password property of that entity.

## Character database

This will save the important variables of the created character to an entity, that uses a save slot as a primary key. The save slot is the username of the user concatenated with a number from 1 to 5. This will create a unique ID to find created characters with. As such the datastore will need a very large number of rows as follows:

- SaveSlot, used to find the correct character when loading a character
- Armour class
- Hit points
- Age
- Alignment

- Background
- Backstory
- Bond
- Character name
- Class
- Equipment
- Eyes
- Features
- Flaw
- Money
- Hair
- Height
- Hit dice
- Ideal
- Initiative
- Languages
- Other proficiencies
- Personality traits
- Real name
- Race
- Skin
- Speed
- Weight
- Spell Ability
- Spell DC
- Spell modifier
- Strength
- Dexterity
- Constitution
- Intelligence
- Wisdom
- Charisma
- Strength modifier
- Dexterity modifier
- Constitution modifier
- Intelligence modifier
- Wisdom modifier
- Charisma modifier
- Strength save
- Dexterity save
- Constitution save
- Intelligence save
- Wisdom save
- Charisma save
- Strength save proficiency//proficiencies need to be saved as they are shown on the character sheet
- Dexterity save proficiency
- Constitution save proficiency

- Intelligence save proficiency
- Wisdom save proficiency
- Charisma save proficiency
- athleticsV
- acrobaticsV
- sleightOfHandV
- stealthV
- arcanaV
- historyV
- investigationV
- natureV
- religionV
- animalHandlingV
- insightV
- medicineV
- perceptionV
- survivalV
- deceptionV
- intimidationV
- performanceV
- persuasionV
- athletics proficiency //proficiencies need to be saved
- acrobatics proficiency
- sleightOfHand proficiency
- stealth proficiency
- arcana proficiency
- history proficiency
- investigation proficiency
- nature proficiency
- religion proficiency
- animalHandling proficiency
- insight proficiency
- medicine proficiency
- perception proficiency
- survival proficiency
- deception proficiency
- intimidation proficiency
- performance proficiency
- persuasion proficiency

## Datastore character functions

The helper class will have a method for adding characters, in order to add all the information to the database, the method will need the character class containing all of the user's choices, their username and the number of the slot they want to save the character to. As such three parameters will be necessary. In order to accept the instance of the character with all the choices as a method the character class will need to be imported  
import characterclass

```
CreateCharacter(usernameIn, saveSlotIn, characterIn){
    Datastore datastore = getDatastoreService
    Entity character = new entity ("character")//create a new entity in the character kind
    character set property ("saveSlot", usernameIn + saveSlotIn)//create the save slot that can be searched to get
    the created character
    character set property ("name", characterIn characterName)
```

```
character set property ("race", characterIn race)
character set property ("background", characterIn Background)
character set property ("class", characterIn class)
//etc for all listed datastore values
datastore put(character)
}
```

There will also be a method for retrieving a created character as a character class object. This will do the opposite, returning a character object, taking the username and save slot as parameters. From the last function the character class is already imported.

```
character getCharacter(usernameIn, saveIn){
character user = new character();
Datastore datastore = getDatastoreService
Filter filter = new filter("saveSlot", check equal, usernameIn + saveIn)//gets the saveslot for the correct user
and save number
Query nameFree = new Query("character") setFilter(filter)
Entity entity = datastore nameFree//now we have the correct character from the database
//each value of the new character class needs to be manually set to the values in the datastore
user characterName = entity getProperty("characterName")
user race = entity getProperty("race")
user background = entity getProperty("background")
user class = entity getProperty("class")
//etc for all other values listed earlier
return user//returns the object with all values from the datastore
}
```

## Trace tables

### Random ability score

```

01     Int Dice[] = {random(1-6), random(1-6), random(1-6), random(1-6)}
02     int y=7
03     Int randScore
04     For( x=0; x<Dice.length; x++){
05         if Dice[x]<y
06         y = Dice[x]
07     }
08     randScore = Dice[0] + Dice[1] + Dice[2] + Dice[3] -y

```

In order to perform a dry run, the random numbers will be generated by real dice rolls. Numbers from one to 6 are generated with a 6 sided dice. The random score should output the 4 highest dice of the four concatenated.

lines	y	Dice[0]	Dice[1]	Dice[2]	Dice[3]	Dice.length	x	randScore	comment
01		2	1	5	4	4			Generate four random numbers from one to 6 in an array
02	7								7 is larger than any dice roll, so y will then be set to the first roll whatever it is
03							0		Initialise the random score to be output
04							0		0 is less than dice length so the for loop runs
05									X = 0, so Dice[0] is checked. 2 is less than 7 so run line 06
06	2								Set y to equal dice[x]
07, 04							1		X is less than dice.length so loop
05									Dice[1] is less than y, so run line 6
06	1								
07, 04							2		X is less than dice.length so loop
05									Dice[2] is greater than y so don't run line 6
07, 04							3		X is less than dice.length so loop
05									Dice[3] is greater than y so don't run line 6

07							4		X is greater than dice.length so don't loop
08								11	Rand score is set to the sum of the scores minus y, as its the same as the lowest dice value.

## Random Heights and weights

Automatically set to human values, the values are dwetermined by setter methods run earlier when the user selects their race. Random numbers will be generated with dice for the dry run

```

01     BaseHeight =56// inches
02     BaseWeight = 110// pounds
03     heightModifier = random(2-16)//two eight sided dice, two to 16
04     weightModifier = random(2-12)//two 6 sided dice, two to 12
05     Height = BaseHeight + heightModifier
06     Weight = BaseWeight + (heightModifier * weightModifier)
07     String HeightImperial = (Height /12) + " " + (Height % 12) + """/convert to an imperial unit string
08     String WeightImperial = (Weight /14) + " stone " + (Weight % 14) + " pounds"

```

lines	BaseHeight	BaseWeight	Height Modifier	Weight Modifier	Height	Weight	Height Imperial	Weight Imperial	comment
01	56								The base height for the human race set in the race setter method
02		110							The base weight for the human race set in the race setter method
03			9						The height modifier for the human race, generated randomly in the human setter method
04				8					The weight modifier for the human race, generated randomly in the human setter method
05					65				The base height plus the height modifier
06						182			The base weight plus the product of the weight modifier and height modifier
07							5'5"		The height is in inches, so it is divided by the number of inches in a foot and

									the remainder is added on in inches
08								13 stone 0 pounds	The weight is in pounds, so it is divided by the number of pounds in a stone and the remainder is added on the end

### Trace table clarification

Many of the algorithms needed simply set values of a variable and require lots of small scale processing. As such the majority of the algorithms do not lend themselves to being written out in trace table form.

## Test Strategy

Much of the testing should occur throughout development regularly running the program and checking basic functionality. However the final version will undergo systematic white box and black box testing to see whether the program outputs to the character sheet correctly and make sure it only allows valid inputs.

The following test strategy is primarily black box. Checking whether inputs give the correct outputs.

<b>Area being tested</b>	<b>Input</b>	<b>type</b>	<b>Expected result</b>
Account creation	Free username and two identical passwords	normal	Being redirected to the login page and the username and password being put in the Datastore
	Taken username and two identical passwords	erroneous	Error message, saying the username is taken
	No username and two identical passwords	erroneous	nothing
	Free username and two different passwords	erroneous	Error message saying the passwords aren't identical
	Free username and one password	erroneous	Error message saying the passwords aren't identical
	Free username and no passwords	erroneous	nothing
	No username and no password	erroneous	nothing
Login	Valid username and password	normal	The form redirects to the first page of character creation
	Valid username and invalid password	erroneous	Error message saying incorrect username or password
	Valid username and no password	erroneous	nothing
	Invalid username and valid password	erroneous	Error message saying incorrect username or password
	No username and valid password	erroneous	Error message saying incorrect username or password
	No username or password	erroneous	nothing
Race page	Submit without making any choices	erroneous	nothing
	Choose a race with other choices and any of those choices missing	erroneous	nothing
	Submit after filling out all the choices for that race	normal	Redirect to the next page
	Choose a race with checkboxes after, such as human variant or half elf and only fill one of the two necessary checkboxes	erroneous	nothing
Name and appearance page	Submit without making any choices	erroneous	nothing
	Submit with all valid values	normal	Redirect to the background page
	Submit with valid data except with erroneous Strings in boxes that accept integers, such as height, weight and age	erroneous	nothing
	Submit with values in the height and	boundary	The random values are

	weight box and the random height and weight checkbox checked		used
	Greater than 11 inches are put in the inches height section	erroneous	nothing
Background page	Submit without making any choices	erroneous	nothing
	Submit with all valid choices needed to be made, made	normal	Redirect to the relevant background page
	Choose a background with further choices and any of those choices not made	erroneous	nothing
	Choose a background with a further checkbox option and only check one option of the two	erroneous	nothing
Point bought ability scores page	Submit without making any choices	erroneous	nothing
	Submit after spending too many points	erroneous	Error message output, saying too many points where spent
	Submit after not spending all your points	erroneous	Error message output, saying too few points where spent
	Submit having spent all your points	normal	Redirect to the class choice page
Saving characters	Save a created character to save slot 1	normal	Character viewable in the character viewer save slot 1 page, with all information outputting to the correct part of the page
	Save a new character over slot 1	normal	The new character overwrites the old, none of the old characters information is shown when
	Save a character to each save slot	normal	Be able to view the correct character from each slot when viewing
Viewing characters	Viewing an average character sheet	normal	All text appears on the character sheet in the correct place
	Viewing a character sheet with as little input as possible i.e. one character Strings	boundary	All text appears on the character sheet in the correct place
	Viewing a character with as much input as possible i.e. max length Strings	boundary	All text appears on the character sheet in the correct place
	View a save slot with nothing saved to it	erroneous	Error message output
Navigation bar	Return to the start of character creation at any point	normal	Choices should be reset, so created characters are normal
	Go to the character viewer at any point	normal	Be able to view all created characters

For the character sheet, for a variety of characters, the values should be output correctly. It will be difficult to optimize this for all platforms so a focus will be put on a 1920 x 1080p monitor, viewing the sheet at 100% view in Google chrome.

The character sheet can only be tested in a black box fashion as it is simply CSS moving JSP outputs around a web page, as such there is no processing occurring under the hood.

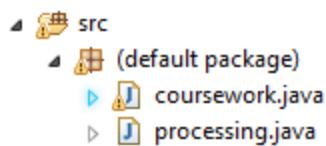
<b>Area being tested</b>	<b>Input</b>	<b>type</b>	<b>Expected result</b>
Main character sheet	Character name, class, Background, Player Name, Race, Alignment and experience points chosen	normal	Output to their appropriate boxes at the top of the character sheet
	Ability scores finalized and chosen	normal	Ability scores output to the small ovals of the ability score section, and their modifiers are output above them in the larger joined squares. Positive modifiers have '+' signs before them, negative ones have a '-'
	Saving throw and skill proficiencies chosen, and skills have been calculated	normal	The Circles showing proficiency are blacked out, and the scores are shown in their blocks. Positive skill scores have '+' signs before them, negative ones have a '-'
	Personality traits, ideals, bonds and flaws are chosen	normal	The information is output to the correct blocks in the top right and line up with the lines provided
	Armour class, initiative, speed, gold, passive perception, hit points and hit dice are chosen	normal	Output to their correct blocks in the top center of the page
	Other proficiencies, equipment, features and traits are chosen	normal	Output to their blocks across multiple lines where needed
Background character sheet	Character name, age, eyes, skin and hair chosen	normal	Output to their appropriate boxes at the top of the character sheet
	Height and weight chosen	normal	Output to their appropriate boxes in imperial units
Spellcasting character sheet	Spellcasting class, spellcasting ability, spell save DC and spell attack bonus chosen	normal	Output to their appropriate boxes at the top of the character sheet
Printing	Print off the sheets	normal	Prints a full page with all the information printed in the correct page

# Software Development

As mentioned in the hardware and software section I Decided to make the game using Java with the eclipse IDE for goggle appengine support. This will allow for easy web development with all the benefits associated with Object-Oriented Programming, for the main class that will use instantiation heavily. The IDE will also allow me to create JS, JSP and CSS files which will be vital to the website.

## Making the main class

I began by creating the main class for holding the character information, this was simply called coursework.java. This is the main class as described earlier that holds instances of all the variables needed to calculate all the values. I also created processing.java to do the important processing tasks.



Initially coursework.java had the variables for character creation added, and then the methods were made. Then I moved on to add the background methods and more variables as they were needed. Also while I was only dealing with one character I used arrays instead of several variables, as no issues would occur due to multiple instances of the class sharing values in arrays.

```
public class coursework {
    static String race;

    String name;
    String eyes;
    String skin;
    String hair;
    int age;
    String size = "medium";

    int height;
    int baseHeight;
    int heightMod;
    int weight;
    int baseWeight;
    int weightMod;

    String background;
    int money;
    String equipment;
    String alignment;
    String[] alignmentTable = { "Chaotic good", "Good", "Lawful good", "Chaotic neutral",
        "Neutral", "Lawful neutral", "Chaotic evil", "Evil", "Lawful evil" };

    static int[] abilityScores = { 8, 8, 8, 8, 8, 8 };// strength, dexterity,
    // constitution, intelligence,
    // wisdom, charisma
    static int[] abilityModifiers = new int[6];

    boolean[] proficiencies = { false, false, false, false, false, false,
        false, false, false, false, false, false, false, false,
        false, false, false };

    /*
     * 0 athletics,1 acrobatics,2 sleightOfHand,3 stealth,4 arcana,5 history, 6
     * investigation,7 nature,8 religion,9 animalHandling,10 insight,11
     * medicine, 12 perception,13 survival,14 deception,15 intimidation,16
     * performance,17 persuasion
     */

    String otherProficiencies;
```

Then setter methods were added in.

```
/*
 * The races
 */

public void HillDwarf(String toolProf) {
    this.race = "Hill Dwarf";
    this.abilityScores[2] = abilityScores[2] + 2; // constitution + 2
    this.speed = 25; // race speed = 25
    this.vision = "Darkvision - you can see in dim light for 60 feet as though it were bright light."
        + " you can see in darkness as though it were dim light";
    this.features = "Dwarven resillience, Stonecunning"; // put in what they
        // do
    this.otherProficiencies = "Battleaxe, Handaxe, Throwing hammer, Warhammer"
        + toolProf;
    this.languages = "Common, Dwarf";
    this.abilityScores[4] = this.abilityScores[4] + 1; // +1 wisdom
    this.features = this.features + ", Dwarven Toughness";
    this.baseHeight = 44;
    this.heightMod = processing.twoD4();
    this.baseWeight = 115;
    this.weightMod = processing.twoD6();
}
```

These were my instancing algorithms planned earlier. I then began making the background setter methods.

```
public void setAlignment(int index){
    this.alignment = alignmentTable[index];
}

/*
 * The backgrounds add personality traits, ideals, bonds and flaws
 */

public void Acolyte(String Language1, String Language2) { // chose two
    // languages
    this.background = "Acolyte";
    this.proficiencies[8] = true;
    this.proficiencies[10] = true;
    this.languages = languages + ", " + Language1 + ", " + Language2;
    this.money = 15;
    this.equipment = "A holy symbol, a prayer book, 5 sticks of incense,"
        + " vestments, common clothes";
    this.features = features + ", Shelter the Faithful";
}
```

I carried on adding variables and setter methods until the character class looks how it does now. At a later point I went back and changed all the integers in the class to long's as Google appspot's Datastore doesn't accept integers. I also converted the arrays for saving choices to variables.

## The Datastore

I then focussed on getting the Datastore to allow the user to save choices, by creating the index and account JSP pages. I had to alter the web.xml file in order to make the URL direct to the index page first:

```
<!-- Default page to serve -->
<welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

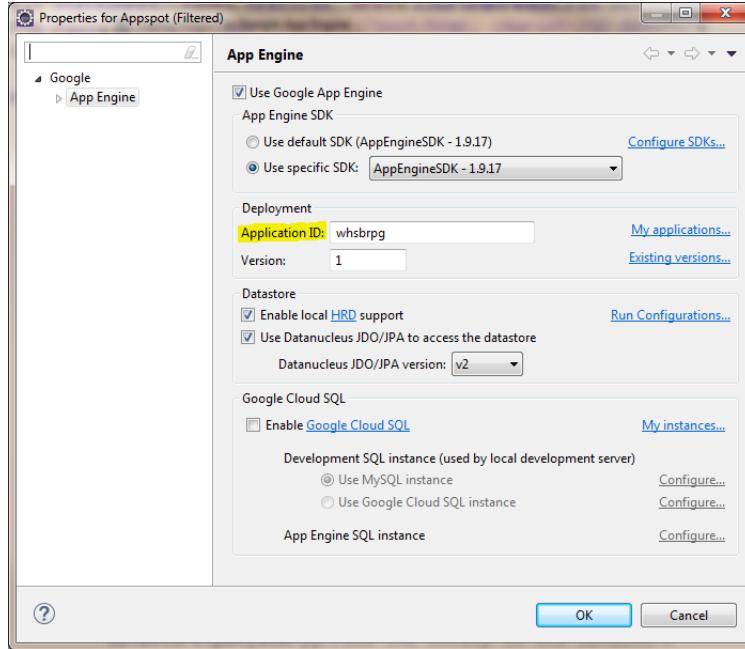
then in the appengine-web.xml file I had to enable session variables, so that users choices could persist between pages.

```
<!-- Session Configuration -->
<sessions-enabled>true</sessions-enabled>
| <!--
```

I also chose a Google appengine URL and chose 'whsbrpg.appspot.com'. I chose this URL as the program would be used within the client's school, and the school doesn't block URLs that begin with whsb.



Having chosen this name, I needed to implement the choice within the eclipse IDE.



Next the helper page was created for its algorithms helping with character creation. Then the first login kind within the Datastore was running.

#### login Entities

ID/Name	password	username
id=5642554087309312	#	#
id=5648554290839552	#	spacegerbil
id=5692462144159744	1	1
id=5730082031140864	YouCanSeeThis	MaxC
id=e70e9999999999999999999999999999	aaaa	ham

The basic instances of the character kind with no choices were also made by this process. As a by-product of using the Datastore GQL could be used to sort through the database when using the Datastore viewer. The character kind:

**character Entities**

ID/Name	owner
id=5189804673204224	spacegerbil

## Character choice JSP pages

Now the Datastore and java backbone of the program was running, all of the player choice pages would need to be created. After logging in the user has their session variable username created, for use later on when

interacting with the Datastore. The java class was imported to allow the object holding the users choice to be created:

```
1 <%@page import="cc.coursework, cc.processing"%>
```

Then when choices were made, the object was set as a session variable to be used by the next page and added to.

An unexpected problem arose when trying to let the user make choices. The race selection was done, however to fulfil the requirements, I needed information about the races to be given and for the next choices for the user to be shown, but in order for the layout to be simple showing all the options at once would be overwhelming, given the little space

## Bram's character creator

### Race

- High elf  Wood elf  Dark elf
- Hill dwarf  Mountain dwarf
- Lightfoot halfling  Stout halfling
- Human  Human (variant)
- Dragonborn
- Forest gnome  Rock gnome
- Half-elf
- Half-orc
- Tiefling

### information:

However the information can't be brought up in a convenient way using JSP and server side processing, so I looked into dynamically creating text and radio buttons with JavaScript.

The solution Involved creating empty div's and running JS functions onclick of the race radio buttons. Then creating labels within the div's and appending them to become radio buttons.

```

var objDiv = document.getElementById("radioDiv");//standard radio button creation with 3 options for tools
var radioItem1 = document.createElement("input");
radioItem1.type = "radio";
radioItem1.name = "tool";
radioItem1.value = "0";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "tool";
radioItem2.value = "1";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "tool";
radioItem3.value = "2";

var objTextNode1 = document.createTextNode("smith's tools");
var objTextNode2 = document.createTextNode("brewer's supplies");
var objTextNode3 = document.createTextNode("mason's tools");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

objDiv.appendChild(objLabel);
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);

```

Unfortunately this solution is very code intensive and requires many lines for little visual outcome. The total radio.JS file ended up taking over 1,400 lines of code. After some research it appears that JQuery would make the process much simpler, however due to time constraints the current solution is pragmatic.

The radio button solution works very well and was surprisingly easy to use with JSP request commands. As you can see in the next screenshot it does the job, creating further information and options onclick, as shown by the high Elf option:

## Bram's character creator

### Race

- High elf  Wood elf  Dark elf
- Hill dwarf  Mountain dwarf
- Lightfoot halfling  Stout halfling
- Human  Human (variant)
- Dragonborn
- Forest gnome  Rock gnome
- Half-elf
- Half-orc
- Tiefling

### choose a language

- Dwarvish  Giant  Gnomish  Goblin  Halfling  Orc  Abyssal  Celestial  Draconic  Deep Speech  Infernal  Primordial  Sylvan  Undercommon

### information:

High elves are the most magically talented of the elves  
they learn an additional language and begin with a magical cantrip. They have improved dexterity and intelligence.

In the JSP switch case block when the user submitted the form, this group of methods was used may times:

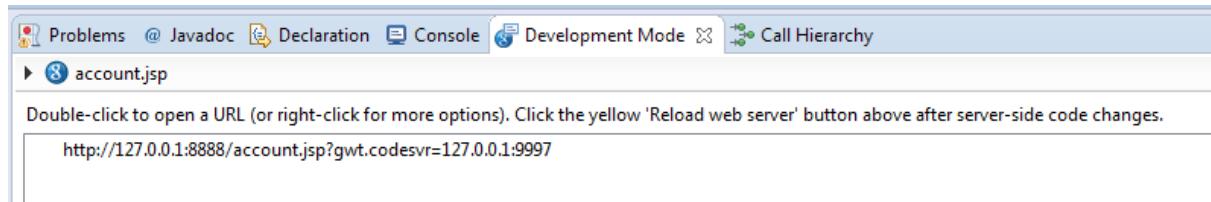
```

player.HighElf(lang); //setter method for high elf race with language parameters
session.setAttribute("player1", player); //set the object of user choices as a session variable for use on later pages
response.sendRedirect("nameAppearance.jsp"); //redirect to the next page for further choices

```

The first setter method would save the users choices to the object from the coursework class. Then the object would be set as a session variable to be retrieved from the next page. Finally the user would be redirected to the next stage of the character creation process.

In order to make sure all the processing had been completed and nothing unnecessary was being saved to the Datastore, characters would only be saved at the very end of character creation. During development the site was run locally. This allowed me to modify the JSP while the website was running in order to test changes. A local version of the Datastore could also be hosted.



The local Datastore was particularly useful for testing:

A screenshot of the Google App Engine Development Console. The 'Datastore Viewer' section is active, showing a list of entities for the 'login' kind. The table includes columns for Key, Write Ops, ID/Name, password, and username. The data shows several entries, each with a unique key and varying values for the other fields.

Key	Write Ops	ID/Name	password	username
agd3aHNlcnBnchlLEgvlsh2dpbhAgjCAGjDCCAw	6	479370697103360	pass	bram
agd3aHNlcnBnchlLEgvlsh2dpbhAgjCAGjCCQw	6	5075345673814016	pass	Bram
agd3aHNlcnBnchlLEgvlsh2dpbhAgjCAGjCICgw	6	5779033115590656	#	#
agd3aHNlcnBnchlLEgvlsh2dpbhAgjCAGjDCgw	6	6060508092301312	q	q
agd3aHNlcnBnchlLEgvlsh2dpbhAgjCAGjCQww	6	6341983069011968	2	2
agd3aHNlcnBnchlLEgvlsh2dpbhAgjCAGjDCCww	6	6482720557367296	b	b

From this point on it was just a matter of adding pages for choices until the character could be saved and the character viewer needed to be implemented.

The final version of the character kind Datastore held every needed piece of information. As shown here:

A screenshot of the Google App Engine Development Console showing the 'character Entities' page. The table lists various character attributes such as AC, HP, acrobatics, alignment, animalHandling, and background. Two specific rows are highlighted: one for a character with ID 5085604337418240 and another for ID 5142291060621312.

ID/Name	AC	HP	acrobatics	acrobaticsb	age	alignment	animalHandling	animalHandlingb	arcana	arcanab	athletics	athleticsb	background	be
<a href="#">id=5085604337418240</a>	15	10	8	True	23	Chaotic neutral	0	False	0	False	1	False	Criminal	*ir
<a href="#">id=5142291060621312</a>	14	10	3	False	31	Good	1	False	0	False	3	True	Soldier	-

When the character viewer needed to be implemented it needed CSS in order to work, so at this point I began developing the CSS for the website as until that point it had been in times new roman, all black with only some aligning on side by side inputs. The majority of the CSS centralised the text, changed the font to sans serif and made titles the pastel purple the client wanted.

The CSS for the character sheet proved much simpler than expected due to the ease of printing HTML pages. All that was required was putting the character sheets as images in the background, importing the object with choices from the Datastore, then outputting the information to a div and using CSS to change the absolute

location of the div so the information was created in the correct area. Line spacing was altered to get the text to correctly spread across multiple lines in certain sections. Finally a print button was added to prompt the user, however using ctrl + p on the image prints a fuller screen version, so it is mentioned to the user on the page.

[click this link, right click and print, or use ctrl + p](#)

The local version and was very useful due to lots of tweaking being necessary for the CSS.

The client was involved frequently during development; we met in person frequently so I could show the progress of the program and make sure it looked and functioned as Mr. Murray wanted. When the local version of the website was complete, he agreed that the final product met what he wanted from the website. Considering that levelling characters would be implemented at a later date.

# Testing

## White box

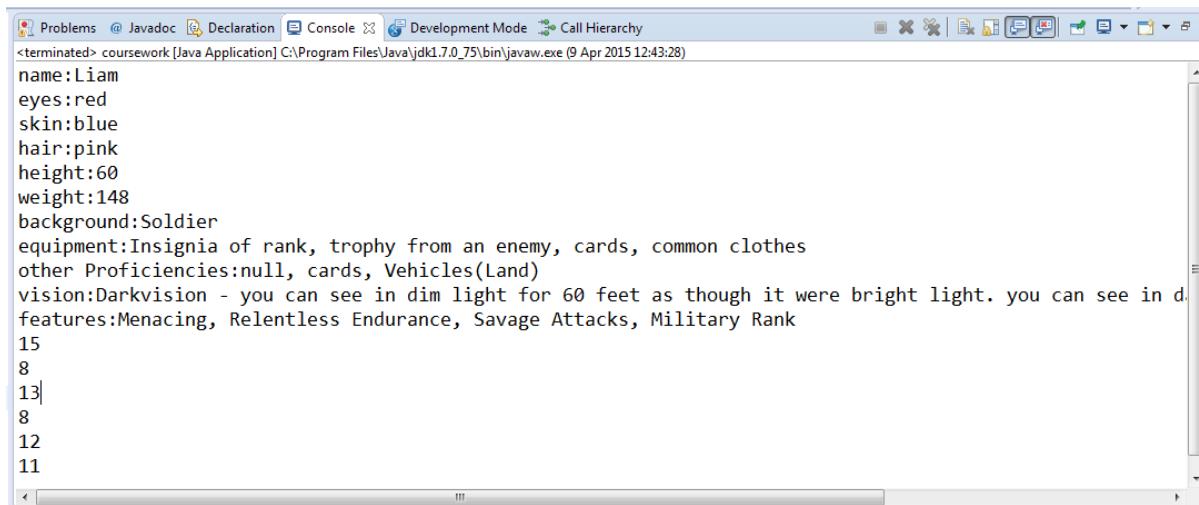
Developing the setters the way I did made testing very easy, as I could manually run the methods in the main method and then check if the values were being displayed correctly with a method that output all the relevant information to the console. This combined with the eclipse IDE stepping feature checking through the program as it ran made white box testing very simple.

```
public static void printAll(coursework in){  
    System.out.println("name:" + in.name );  
    System.out.println("eyes:" + in.eyes );  
    System.out.println("skin:" + in.skin );  
    System.out.println("hair:" + in.hair );  
    System.out.println("height:" + in.height );  
    System.out.println("weight:" + in.weight );  
    System.out.println("background:" + in.background );  
    System.out.println("equipment:" + in.equipment );  
    System.out.println("other Proficiencies:" + in.otherProficiencies );  
    System.out.println("vision:" + in.vision );  
    System.out.println("features:" + in.features );  
    for(int y=0; y<6; y++){  
        System.out.println(abilityScores[y]);  
    }  
}
```

In the main method I would simply instance the class, and then run the methods as I added them to see if the correct values were being output.

```
public static void main(String args[]){  
    coursework Liam = new coursework();  
    Liam.HalfOrc();  
    Liam.Background("Liam", "red", "blue", "pink", 13, true, 1, 1 );  
    Liam.Soldier("cards");  
    randAbility();  
    printAll(Liam);  
}
```

This would output to the console like this for easy error checking:



The screenshot shows the Eclipse IDE's Console view. The title bar includes tabs for Problems, Javadoc, Declaration, Console, Development Mode, and Call Hierarchy. The console output window displays the following text:  
name:Liam  
eyes:red  
skin:blue  
hair:pink  
height:60  
weight:148  
background:Soldier  
equipment:Insignia of rank, trophy from an enemy, cards, common clothes  
other Proficiencies:null, cards, Vehicles(Land)  
vision:Darkvision - you can see in dim light for 60 feet as though it were bright light. you can see in d  
features:Menacing, Relentless Endurance, Savage Attacks, Military Rank  
15  
8  
13|  
8  
12  
11

I continually tested the program as features were implemented. This meant by the time the program was almost finished, most of the errors had been sorted out.

## Black Box testing

### Main pages

Area being tested	Input	Expected result	Actual result	Relevant screenshot
Account creation	Free username and two identical passwords	Being redirected to the login page and the username and password being put in the Datastore	Expected result	1, 2
	Taken username and two identical passwords	Error message, saying the username is taken	Expected result	3
	No username and two identical passwords	nothing	Expected result	
	Free username and two different passwords	Error message saying the passwords aren't identical	Expected result	4
	Free username and one password	Error message saying the passwords aren't identical	nothing	
	Free username and no passwords	nothing	Expected result	
	No username and no password	nothing	Expected result	
Login	Valid username and password	The form redirects to the first page of character creation	Expected result	
	Valid username and invalid password	Error message saying incorrect username or password	Expected result	5
	Valid username and no password	nothing	Expected result	
	Invalid username and valid password	Error message saying incorrect username or password	Expected result	6
	No username and valid password	Error message saying incorrect username or password	Expected result	7
	No username or password	nothing	Expected result	
Race page	Submit without making any choices	nothing	Expected result	
	Choose a race with other choices and any of those choices missing	nothing	Expected result	
	Submit after filling out all the choices for that race	Redirect to the next page	Expected result	
	Choose a race with checkboxes after, such as human variant or half elf and only fill one of the two	nothing	Redirects regardless	This error has now been fixed by rearranging its try catch block, it now

	necessary checkboxes			gives the Expected result
Name and appearance page	Submit without making any choices	nothing	Expected result	
	Submit with all valid values	Redirect to the background page	Expected result	
	Submit with valid data except with erogenous Strings in boxes that accept integers, such as height, weight and age	nothing	Expected result	8
	Submit with values in the height and weight box and the random height and weight checkbox checked	The random values are used	Expected result, as shown by the screenshot, the random value was used	9, 10
	Greater than 11 inches are put in the inches height section	nothing	Expected result	
Background page	Submit without making any choices	nothing	Expected result	
	Submit with all valid choices needed to be made, made	Redirect to the relevant background page	Expected result	
	Choose a background with further choices and any of those choices not made	nothing	Expected result	
	Choose a background with a further checkbox option and only check one option of the two	nothing	Expected result	
Point bought ability scores page	Submit without making any choices	nothing	Expected result	
	Submit after spending too many points	Error message output, saying too many points where spent	Expected result	11
	Submit after not spending all your points	Error message output, saying too few points where spent	Expected result	12
	Submit having spent all your points	Redirect to the class choice page	Expected result	
Saving characters	Save a created character to save slot 1	Character viewable in the character viewer save slot 1 page	Expected result	13, 14
	Save a new character over slot 1	The new character overwrites the old, none of the old characters information is shown	Expected result	15, 16
	Save a character to each save slot	Be able to view the correct character from	Expected result	

		each slot when viewing		
Viewing characters	Viewing a character sheet with as little input as possible i.e. one character Strings	All text appears on the character sheet in the correct place	Expected result	17, 18
	Viewing a character with as much input as possible I.e. max length Strings	All text appears on the character sheet in the correct place	The text inputs overflowed	19
	View a save slot with nothing saved to it	Error message output	The page redirects to a blank character sheet	
Navigation bar	Return to the start of character creation at any point	Choices should be reset, so created characters are normal	Expected result	
	Go to the character viewer at any point	Be able to view all created characters	Expected result	

Number	Screenshot						
1	<p style="text-align: center;"><b>create account</b></p> <p style="text-align: center;">Username: <input type="text" value="arnold"/></p> <p style="text-align: center;">Password: <input type="password" value="....."/></p> <p style="text-align: center;">Re-enter Password: <input type="password" value="....."/></p> <p style="text-align: center;"><input type="button" value="Submit"/></p>						
2	<p>Query Create</p> <p>By kind: login kinds as of 0:00:53 ago</p> <p>Options</p> <p>By GQL: <input type="text" value="SELECT * FROM login WHERE username = 'arnold'"/></p> <p>Run Query</p> <p><b>login Entities</b></p> <table border="1"> <thead> <tr> <th>ID/Name</th> <th>password</th> <th>username</th> </tr> </thead> <tbody> <tr> <td><a href="#">id=5714163003293696</a></td> <td>starscream</td> <td>arnold</td> </tr> </tbody> </table>	ID/Name	password	username	<a href="#">id=5714163003293696</a>	starscream	arnold
ID/Name	password	username					
<a href="#">id=5714163003293696</a>	starscream	arnold					
3	The username: arnold is taken						
4	your passwords are not identical						
5	your username or password is incorrect						
6	your username or password is incorrect						
7	your username or password is incorrect						

8

## Bram's character creator

### Name and Appearance

character's name:

your name:

Age: nineteen-eighty four

Height  foot  inches

Eyes: none

random height and weight

Skin: none

Weight  pounds

Hair: none

9

Height  foot  inches

random height and weight

Weight  pounds

10

height: 5' 5" weight: 9 stone 0 pounds

11

you have spent too many points

12

you still have points left

13

## Bram's character creator

### Character Viewer

save slot 1  save slot 2  save slot 3  save slot 4  save slot 5

main sheet  background sheet  spell sheet

14

**DUNGEONS & DRAGONS®**

**CHARACTER NAME:** Jack Grimfire

<b>CLASS &amp; LEVEL:</b>	<b>RACE:</b>	<b>BACKGROUND:</b>	<b>PLAYER NAME:</b>
Barbarian	Dragonborn	Soldier	Bram
Chaotic neutral	0	ALIGNMENT	EXPERIENCE POINTS

**STRENGTH**: +4 (18)    **INSPIRATION**:    **PROFICIENCY BONUS**: +2

**ARMOR CLASS**: 14    **INITIATIVE**: +2    **SPEED**: 30  
Hit Point Maximum: 14

**PERSONALITY TRAITS:** I can stare down a hell hound without flinching.

15

## Bram's character creator

### Character Viewer

save slot 1  save slot 2  save slot 3  save slot 4  save slot 5

main sheet  background sheet  spell sheet

**Submit**

16

**DUNGEONS & DRAGONS®**

**CHARACTER NAME:** Eric

<b>CLASS &amp; LEVEL:</b>	<b>RACE:</b>	<b>BACKGROUND:</b>	<b>PLAYER NAME:</b>
Sorcerer	Forest Gnome	Hermit	Bram
Chaotic good	0	ALIGNMENT	EXPERIENCE POINTS

**STRENGTH**: -1 (8)    **INSPIRATION**:    **PROFICIENCY BONUS**: +2

**ARMOR CLASS**: 9    **INITIATIVE**: -1    **SPEED**: 25  
Hit Point Maximum: 8

**PERSONALITY TRAITS:** I often get lost in my own thoughts and contemplation, becoming oblivious to my surroundings.

**DUNGEONS & DRAGONS®**

1 CHARACTER NAME	Wizard CLASS & LEVEL <b>Tiefling</b> RACE	Urchin BACKGROUND <b>Evil</b> ALIGNMENT	1 PLAYER NAME 0 EXPERIENCE POINTS
<b>STRENGTH</b> -3 5  <b>DEXTERITY</b> -2 6  <b>CONSTITUTION</b> -2 7  <b>INTELLIGENCE</b> -1 9  <b>WISDOM</b> 0 10  <b>CHARISMA</b> +1 13	<b>INSPIRATION</b>  <b>PROFICIENCY BONUS</b> +2	<b>ARMOR CLASS</b> 8  <b>INITIATIVE</b> -2  <b>SPEED</b> 30	<b>PERSONALITY TRAITS</b>  <b>IDEALS</b>  <b>BONDS</b>  <b>FLAWS</b>
	<b>HIT POINT MAXIMUM</b> 4  <b>CURRENT HIT POINTS</b>  <b>TEMPORARY HIT POINTS</b>	<b>HIT DICE</b> Total 1D6  <b>SUCCESES</b> ○○○ <b>FAILURES</b> ○○○ <b>DEATH SAVES</b>	<b>SKILLS</b>  ○ -2 Acrobatics (Dex) ○ 0 Animal Handling (Wis) ● +1 Arcana (Int) ○ -3 Athletics (Str) ○ +1 Deception (Cha) ● +1 History (Int) ○ 0 Insight (Wis) ○ +1 Intimidation (Cha) ○ -1 Investigation (Int) ○ 0 Medicine (Wis) ○ -1 Nature (Int) ○ 0 Perception (Wis) ○ +1 Performance (Cha) ○ +1 Persuasion (Cha) ○ -1 Religion (Int) ● 0 Sleight of Hand (Dex) ● 0 Stealth (Dex) ○ 0 Survival (Wis)
<b>PASSIVE WISDOM (PERCEPTION)</b> 10  disguise kit, thieves tools, daggers, darts, slings, quarterstaffs, light crossbows Common, Infernal	A small knife, a map of the city you grew up in, a pet mouse, a token to remember your parents by, a set of common clothes, , quarterstaff, component pouch, scholar's pack, spellbook		

Darkvision - you can see in dim light for 60 feet as though it were bright light. You can see in darkness as though it were dim light  
 Hellish Resistance, Infernal Legacy, City Secrets, Spellcasting, Arcane Recovery

**D&D®**

1 CHARACTER NAME	1 AGE EYES	1' 1" HEIGHT 1 SKIN	0 stone 1 pounds WEIGHT 1 HAIR
---------------------	------------------	------------------------------	---

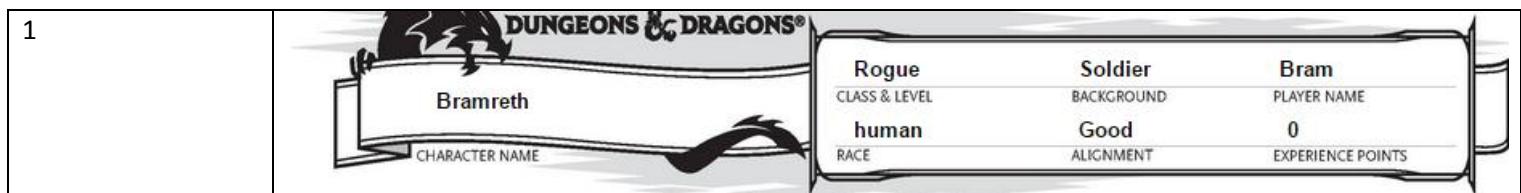
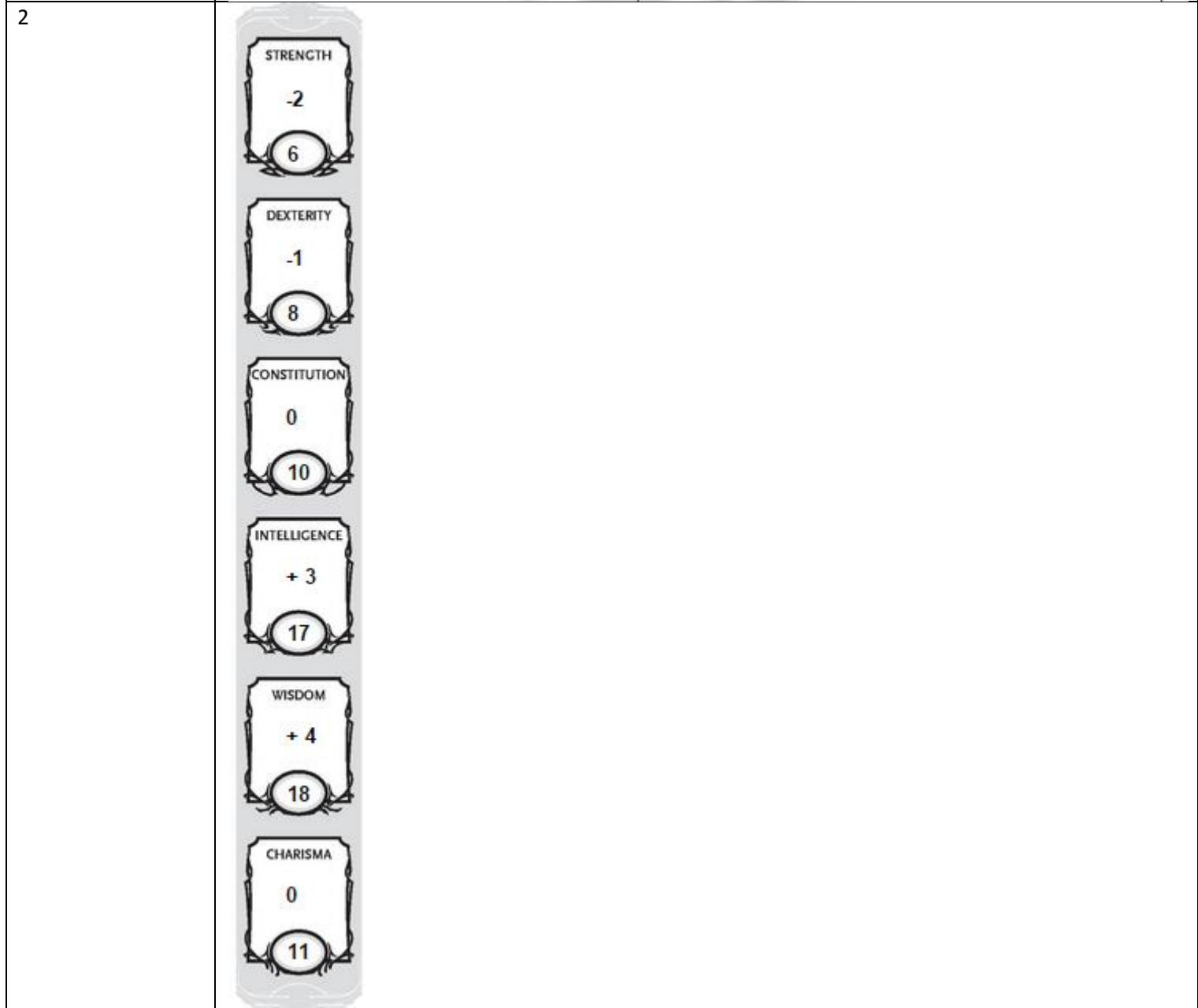


## Character sheet pages

Area being tested	Input	Expected result	Actual result	Relevant screenshot
Main character sheet	Character name, class, Background, Player Name, Race, Alignment and experience points chosen	Output to their appropriate boxes at the top of the character sheet	Expected result	1
	Ability scores finalized and chosen	Ability scores output to the small ovals of the ability score section, and their modifiers are output above them in the larger joined squares. Positive modifiers have '+' signs before them, negative ones have a '-'	Expected result	2
	Saving throw and skill proficiencies chosen, and skills have been calculated	The Circles showing proficiency are blacked out, and the scores are shown in their blocks. Positive skill scores have '+' signs before them, negative ones have a '-'	Expected result	3
	Personality traits, ideals, bonds and flaws are chosen	The information is output to the correct blocks in the top right and line up with the lines provided	Expected result	4
	Armour class, initiative, speed, passive perception, hit points and hit dice are chosen	Output to their correct blocks in the top center of the page	Expected result	5, 6
	Other proficiencies and languages, equipment and gold, features and traits are chosen	Output to their blocks across multiple lines where needed	Expected result	7, 8, 9
Background character sheet	Character name, age, eyes, skin and hair chosen	Output to their appropriate boxes at the top of the character sheet	Expected result	10
	Height and weight chosen	Output to their appropriate boxes in imperial units	Expected result	11
Spellcasting character sheet	Spellcasting class, spellcasting ability, spell save DC and spell attack bonus chosen	Output to their appropriate boxes at the top of the character sheet	Expected result	12

Printing	Print off the sheets	Prints a full page with all the information printed in the correct page	Expected result	13
----------	----------------------	---	-----------------	----

These screenshots were each taken from different character sheets, so that I could make sure the values were consistently outputting correctly.

1	 <p>DUNGEONS &amp; DRAGONS®</p> <p>Bramreth</p> <p>CHARACTER NAME</p> <table border="1"> <tr> <td>Rogue</td> <td>Soldier</td> <td>Bram</td> </tr> <tr> <td>CLASS &amp; LEVEL</td> <td>BACKGROUND</td> <td>PLAYER NAME</td> </tr> <tr> <td>human</td> <td>Good</td> <td>0</td> </tr> <tr> <td>RACE</td> <td>ALIGNMENT</td> <td>EXPERIENCE POINTS</td> </tr> </table>				Rogue	Soldier	Bram	CLASS & LEVEL	BACKGROUND	PLAYER NAME	human	Good	0	RACE	ALIGNMENT	EXPERIENCE POINTS
Rogue	Soldier	Bram														
CLASS & LEVEL	BACKGROUND	PLAYER NAME														
human	Good	0														
RACE	ALIGNMENT	EXPERIENCE POINTS														
2	 <table border="1"> <tr> <td>STRENGTH -2 6</td> <td>DEXTERITY -1 8</td> <td>CONSTITUTION 0 10</td> <td>INTELLIGENCE + 3 17</td> </tr> <tr> <td>WISDOM + 4 18</td> <td>CHARISMA 0 11</td> <td></td> <td></td> </tr> </table>				STRENGTH -2 6	DEXTERITY -1 8	CONSTITUTION 0 10	INTELLIGENCE + 3 17	WISDOM + 4 18	CHARISMA 0 11						
STRENGTH -2 6	DEXTERITY -1 8	CONSTITUTION 0 10	INTELLIGENCE + 3 17													
WISDOM + 4 18	CHARISMA 0 11															

3

○	<u>-2</u>	Strength
○	<u>-1</u>	Dexterity
○	<u>0</u>	Constitution
○	<u>+3</u>	Intelligence
●	<u>+6</u>	Wisdom
●	<u>+2</u>	Charisma

## SAVING THROWS

○	<u>-1</u>	Acrobatics (Dex)
○	<u>+4</u>	Animal Handling (Wis)
●	<u>+5</u>	Arcana (Int)
○	<u>-2</u>	Athletics (Str)
○	<u>0</u>	Deception (Cha)
●	<u>+5</u>	History (Int)
●	<u>+6</u>	Insight (Wis)
○	<u>0</u>	Intimidation (Cha)
○	<u>+3</u>	Investigation (Int)
●	<u>+6</u>	Medicine (Wis)
●	<u>+7</u>	Nature (Int)
○	<u>+4</u>	Perception (Wis)
○	<u>0</u>	Performance (Cha)
●	<u>+2</u>	Persuasion (Cha)
●	<u>+7</u>	Religion (Int)
○	<u>-1</u>	Sleight of Hand (Dex)
○	<u>-1</u>	Stealth (Dex)
○	<u>+4</u>	Survival (Wis)

## SKILLS

4

I use polysyllabic words that convey the impression of great erudition.

## PERSONALITY TRAITS

Knowledge. The path to power and self-improvement is through knowledge.

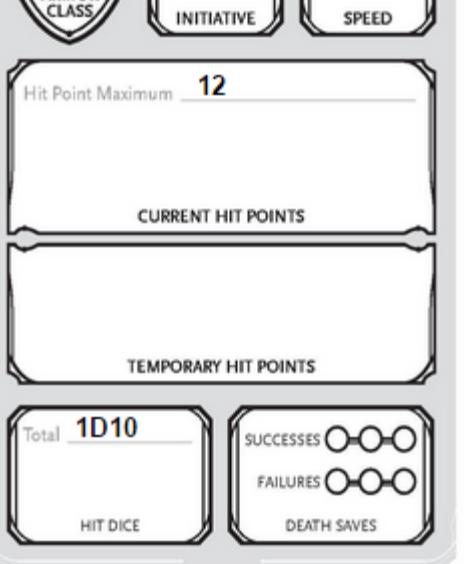
## IDEALS

I work to preserve a library, university, scriptorium or monastary.

## BONDS

Most people scream and run when they see a demon. I stop and take notes on its anatomy.

## FLAWS

5	 <p>Hit Point Maximum <u>12</u></p> <p>CURRENT HIT POINTS</p> <p>TEMPORARY HIT POINTS</p> <p>Total <u>1D10</u></p> <p>HIT DICE</p> <p>SUCCESES  FAILURES </p> <p>DEATH SAVES</p>
6	 <p>PASSIVE WISDOM (PERCEPTION) <u>11</u></p>
7	 <p>light armour, medium armour, shields, all simple weapons Common, Dwarvish, Elvish, Gnomish, Goblin, Orc</p> <p>OTHER PROFICIENCIES &amp; LANGUAGES</p>

8

Darkvision - you can see in dim light for 60 feet  
 as though it were bright light. You can see in darkness as though it were dim light  
 Menacing, Relentless Endurance, Savage Attacks, Discovery, Spellcasting, Arcane Recovery

## FEATURES &amp; TRAITS

9

CP Insignia of rank, trophy from an enemy, Dice set, common clothes, , rapier, shortsword, explorer's pack, leather armour, two daggers, thief's tools

10

## EQUIPMENT

10

Mark Schenker

CHARACTER NAME

37

AGE

Brown

EYES

5' 6"

HEIGHT

Dark green

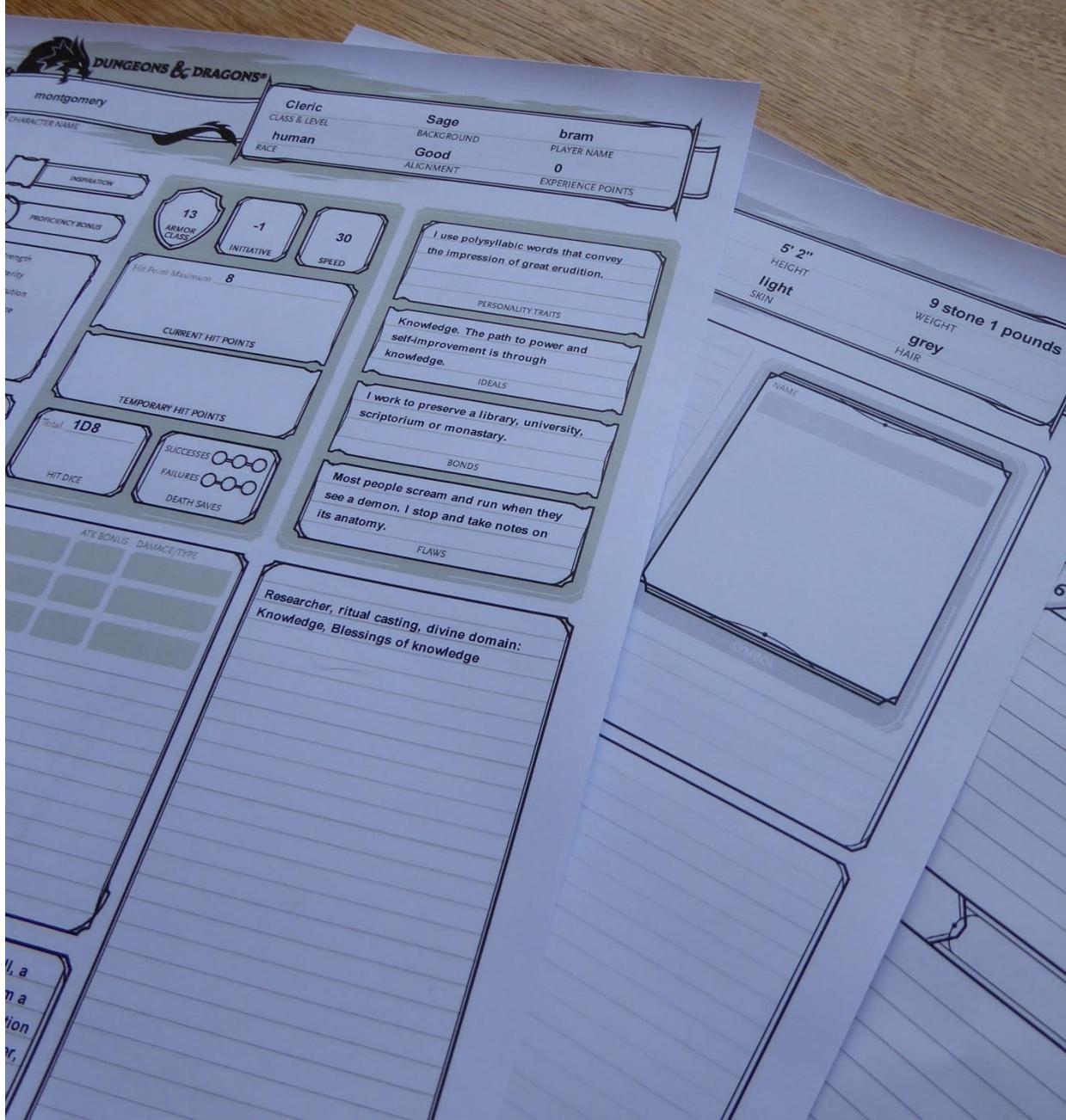
SKIN

11 stone 10 pounds

WEIGHT

Black

HAIR

11	<b>6' 7"</b> HEIGHT	<b>17 stone 5 pounds</b> WEIGHT
12		
13		

## Results of testing

Using my test plan I have shown that my solution works successfully with only a few bugs.

The checkbox error was a simple logic error to do with the try catch block being within the method rather than outside, allowing the code to continue after the error. This code, needed to be re-organised to

```
if (request.getParameter("abils") != null) {
    try{
String[] a = request.getParameterValues("abils");//get all choices from ability
score checkboxes
abil1 = Integer.parseInt(a[0]);//set first ability score number as first value in
array
abil2 = Integer.parseInt(a[1]);//set second ability score as second value in array
    }catch(Exception e){//due to js there can only be two choices
    }
}
if (request.getParameter("prof") != null) {//get all proficiency choices
    try{
String[] b = request.getParameterValues("profs");
prof1 = Integer.parseInt(b[0]);//set profession 1 to first selected checkbox
prof2 = Integer.parseInt(b[1]);//set profession 2 to second checkbox
    }catch(Exception e){
    }
player.reset();
player.HalfElf(abil1, abil2, prof1, prof2, lang);//set to half elf with all
parameters
session.setAttribute("player1", player);
response.sendRedirect("nameAppearance.jsp");
}
}
}
break;
```

needed to be re-arranged to:

```
try{
    if (request.getParameter("abils") != null) {
        String[] a = request.getParameterValues("abils");//get all choices
from ability score checkboxes
        abil1 = Integer.parseInt(a[0]);//set first ability score number as
first value in array
        abil2 = Integer.parseInt(a[1]);//set second ability score as second
value in array
        if (request.getParameter("profs") != null) {//get all proficiency
choices
            String[] b = request.getParameterValues("profs");
            prof1 = Integer.parseInt(b[0]);//set profession 1 to first
selected checkbox
            prof2 = Integer.parseInt(b[1]);//set profession 2 to second
checkbox
            player.reset();
            player.HalfElf(abil1, abil2, prof1, prof2, lang);//set to half
elf with all parameters
            session.setAttribute("player1", player);
            response.sendRedirect("nameAppearance.jsp");
        }
    }
}
```

```
}catch(Exception e){  
}  
}  
  
break;
```

The logic error was almost identical for the human variant.

The error with text being entered overflowing was very easily solved by adding `maxlength = "17"` to the text inputs, as it was at 17 characters they overflowed, so limiting the number of characters that could be input stopped the problem.

## User testing

Throughout development family and friends had been testing the system and creating characters, this helped to find many small grammatical errors and problems with saving incorrect information to character sheets at the end. The users found one more error that the test plan didn't accommodate for, when the user sets their ability scores. If the user sets their ability score then uses the back button to return to the ability score page, the values they choose are added on top of their already chosen values, this means users can get their ability scores much higher than is allowed in the rules:

The screenshot shows a web-based RPG character creator interface. At the top, there's a navigation bar with links for 'Character Creator' and 'Character Viewer'. Below the navigation, the title 'Bram's character creator' is displayed, followed by 'Choose a class'. A horizontal bar lists various character classes: Barbarian, Bard, Cleric, Druid, Fighter, Monk, Paladin, Ranger, Rogue, Sorceror, Warlock, and Wizard. Underneath this, a section titled 'class information:' contains a 'Submit' button. To the left, there's a sidebar with ability scores: Strength (+14, total 38), Dexterity (+14, total 38), Constitution (+14, total 38), Intelligence (+11, total 32), Wisdom (+10, total 31), and Charisma (+11, total 33). Each ability score has a corresponding 'INSPIRATION' box and a 'PROFICIENCY BONUS' box. To the right of the sidebar, there are two columns of skill lists under 'SAVING THROWS' and 'SKILLS'. The 'SAVING THROWS' column includes: +14 Strength, +14 Dexterity, +14 Constitution, +13 Intelligence, +12 Wisdom, and +11 Charisma. The 'SKILLS' column includes: +14 Acrobatics (Dex), +10 Animal Handling (Wis), +13 Arcana (Int), +14 Athletics (Str), +11 Deception (Cha), +13 History (Int), +10 Insight (Wis), +11 Intimidation (Cha), +11 Investigation (Int), +10 Medicine (Wis), +11 Nature (Int), +10 Perception (Wis), +11 Performance (Cha), +11 Persuasion (Cha), +11 Religion (Int), +16 Sleight of Hand (Dex), +16 Stealth (Dex), and +10 Survival (Wis).

Ability scores are not allowed to go above 20 under any circumstances, and this can happen if a user wants to go back and change their choice. Unfortunately due to the way the back button works, session variables don't update properly and even when I introduce Boolean gates to stop the user returning to the page, they don't refresh and the user can still add more to their score. This will require more research to solve.

Some characters created by users will be shown here, aside from the client, the members of the club where very helpful for testing.

### Max (club member)

By GQL: `SELECT * FROM character WHERE realName = 'Max'`

Learn more about [GQL syntax](#).

#### Character Entities in Empty Namespace

Prev 20 1-1 Next 20

ID/Name	AC	HP	acrobatics	acrobaticsb	age	alignment	animalHandling	animalHandlingb	arcana	arcanab	athletics	athleticsb	background	backstory	
<a href="#">id=5085604337418240</a>	15	10	8		True	23	Chaotic neutral	0		False	0	False	1	False	Criminal *insert generic test details here*

### Mr. Murray

The client did a test run to see how the program was coming along.

By GQL: `SELECT * FROM character WHERE realName = 'Mister MurRay'`

Learn more about [GQL syntax](#).

#### Character Entities

Prev 20 1-1 Next 20

ID/Name	AC	HP	acrobatics	acrobaticsb	age	alignment	animalHandling	animalHandlingb	arcana	arcanab	athletics	athleticsb	background		
<a href="#">id=5649050225344512</a>	27	27	16		False	1337	Neutral	10		False	12	False	22	True	Guild Merchant

### Dave (club member)

By GQL: `SELECT * FROM character WHERE realName = 'HowLongCanTheseUsernamesBe? Probably very long! oh my I wonder if they take up no space like filenames haha that would be cool, err what was I'`

Learn more about [GQL syntax](#).

#### Character Entities

Prev 20 1-1 Next 20

ID/Name	AC	HP	acrobatics	acrobaticsb	age	alignment	animalHandling	animalHandlingb	arcana	arcanab	athletics	athleticsb	background	backstory	
<a href="#">id=5712536552865792</a>	755	345	412		False	0	Chaotic evil	125		False	229	False	439	True	Soldier ~@!?!\$!%? \$!@?\$_! ***voices from the abyss*** ~~)!?!\$? \$!&&

## Connor (club member)

By GQL: `SELECT * FROM character WHERE realName = '1'`

Learn more about [GQL syntax](#).

### Character Entities

< Prev 20 1-3 Next 20 >														
<input type="checkbox"/> ID/Name	AC	HP	acrobatics	acrobaticsb	age	alignment	animalHandling	animalHandlingb	arcana	arcanab	athletics	athleticsb	background	backstory

[id=5656058538229760](#) 24 20 14 False 1 Neutral 10 False 13 True 14 False Urchin 1

[id=5687539843203072](#) 10 8 0 False 1 Neutral 1 False 1 False 1 False Urchin 1

[id=5742796208078848](#) 12 9 2 True 1 Lawful neutral 4 True 2 False 1 True Folk Hero ekl;srghuoerhg

After testing with the test plan and test users had been completed, all that was left was to show the final form of the program to the client. This is continued in the evaluation section.

## Evaluation

I can conclude that the final system meets almost all of the requirements specification, falling short in four sections, all of which are features that can be added later in the websites lifespan. The solution meets almost all requirements and fulfils its primary object of creating a fast alternative for creating characters.

### Input requirements

reference	input requirements	Why it was met
1.a.	users should be able to log in to an account	The main URL links to the login page, where after an account is created the user can log in.
1.b.	users should be able to create multiple characters	On character finalisation users are presented with 5 save slots allowing them to create multiple characters.
1.c.i.	users should be able to choose their race	The first page on login allows the user to select their characters race and further sub choices if there are any.
1.c.ii.	users should be able to choose their appearance	After race choice is a page exclusively for choosing names and appearance.
1.c.iii	users should be able to choose their background	After name and appearance choices the user is redirected to a page for background choices with sub choices and then they get to choose their background specific personality choices.
1.c.iv	users should be able to choose their ability scores	After personality the user gets to choose how their ability choices are allocated, then are directed to the appropriate page for score allocation.
1.c.v	users should be able to choose their class	After ability scores are decided the user is redirected to a class overview page and can then choose a class and all sub choices for that class.
1.c.vi	users should be able to choose all subsequent choices within those listed above	Major choices bring up dynamic radio buttons for sub choices
1.d.	users should be able to choose created characters to view	In the character viewer, users can select a save slot they want and what page they want to view from that save slot.
1.e.	users should be able to increase their characters experience points	After E-mails with the client they agreed that the character leveller would not be able to be implemented in the timeframe, and the feature was not integral to the product.
1.f.	users should be able to edit and delete created characters	This objective was not met, however the ability to overwrite created characters and the ease of character creation makes the feature less necessary.

1.g.	users should be able to use their mouse to select options	All web pages are primarily mouse driven.
1.h.	users should be able to save their character sheets, to access whenever they need to	The characters are saved to the Datastore, and can be accessed whenever the user logs back on to the account.

## Processing requirements

reference	Processing requirements	Why it was met
2.a.	The program should provide information about the choices the user makes	Main choices have text about the choice be dynamically created, for races, backgrounds and classes.
2.b.	The program should recommend choices for classes	This objective was not met, however it is an ease of use function, not relevant to all users, that can be implemented at a later date.
2.c.	The program should be able to create characters and obtain all the appropriate information for choices	Characters are created fully, the algorithms produce the correct outputs in all tested cases.
2.d.	The program should allow Email and password recovery on login	This objective was not met due to complications of implementing tying E-mails to accounts and verification involved, little would be gained for the user, against the ease of account creation currently implemented.
2.e.	The program should store created characters for future access	Created characters on finalisation are saved to the Datastore and can be accessed at a later point.
2.f.	The program should be able to generate random values when wanted e.g. ability scores	Random values are available for all random scores described in the players guide.
2.g.	The program should assimilate all the information in one easy to use document, if possible, a character sheet.	A fully functional set of character sheets is created for printing in the character viewer.
2.h.i.	The program should calculate all Ability scores	Ability scores are chosen by the user and the race choice adds relevant bonuses.
2.h.ii.	The program should calculate all Ability modifiers	Ability modifiers are calculated from finalised ability scores.
2.h.iii.	The program should calculate all Saving throws	The class choice determines saving throw proficiency, then after ability modifier calculation and proficiency modifier addition, the saving throws are calculated
2.h.iv.	The program should calculate all Skill modifiers	All skill proficiencies are saved, the relevant ability score is used and the proficiency bonuses are added appropriately. Choices that double certain proficiency bonuses such as rogue expertise are accounted for.
2.h.v.	The program should calculate Armour class	All armours are saved on class choice as well as shields and features, then at the end of creation the AC is calculated depending on the armour.
2.h.vi.	The program should calculate Initiative	Initiative is calculated after the dexterity modifier.
2.h.vii.	The program should calculate Hit points	This is calculated on class choice, as it depends on both class and constitution modifier, which will not

		change after class choice.
2.h.viii.	The program should calculate Hit dice	This is set on class choice.
2.h.ix.	The program should calculate Passive perception	This is calculated after skill scores, taking the value of the players perception skill.

## Design requirements

reference	Design requirements	Why it was met
3.a.	The layout should be minimalistic and easy to use	Almost all choices use radio buttons, making choices relatively simple, and as streamlined as possible. Only relevant information is displayed, decreasing clutter and showing only what the user needs to know.
3.b.	there should be a login screen first then a main menu with the three main options presented	This was met, however as previously discussed, the client agreed the third page for character levelling wasn't necessary, so only the character creator and viewer pages are presented.
3.c.	when creating characters the screen should bring up further choices	Dynamic radio buttons bring up sub choices, when a user makes a main choices such as race or backgrounds. Classes needed their own page for all the choices.
3.d.	Left to right, top down familiar design	Main pages are presented in the top left. Then information is presented from the top to the bottom of the page.
3.e.	use of complementary colours	No contrasting colours were used. Only shades of grey and the pastel purple the client wanted.
3.f.	links to the main pages should be available on any part of the website in a bar at the top	The navigation bar is present in the top left on every page, to create a new character and view existing ones.
3.g.	Centralised design	Information on every page was centralised except for when there were two columns which would be side by side in the centre, such as ability score choices.
3.h.	Use of Pastel purple	Pastel purple was used for titles and headers on every page as per the clients request.

## Output requirements

reference	Output requirements	Why it was met
4.a.	Output all the information to the correct boxes in the character sheet	As shown in testing all information outputs correctly on the webpage and in printed character sheets.
4.b.	The program should be able to output all 3 completed character sheets for the user to view	Each page can be selected in the character viewer, printed off and viewed separately.
4.c.	The program should be able to print it off character sheets with all the needed information	A print link is shown on each sheet page, as well as instructions for printing off with hotkeys or when right clicking.
4.d.	output all created characters to the GM so they can see what the players characters are	This objective was not met, user types were not implemented. Instead as character sheets are printable, the GM can actually see physical copies if needed.

## User response

Dear Liam,

After extensive testing The project is complete, could we meet anytime soon for you to give your thoughts on the final product?

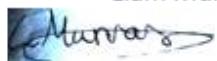
kind regards,  
Bram

Hello Bram,

That's good to hear, I can meet you from 10:30am to 11:15am tomorrow, does that work for you?

Yours faithfully,

Liam Murray



Dear Liam,

Sure thing, I'll meet you in the 6th form basement.

Kind regards,  
Bram

At the meeting I will ask Mr. Murray to fill out the requirements specification below, as well as his thoughts on the results of testing, so he can rate how well he feels I have met the objectives I was given. I will also bring a series of questions, on his opinions of the program, for him to sign at the end.

## Input success

Input requirements	Client success rating
users should be able to log in to an account	10/10
users should be able to create multiple characters	10/10
users should be able to choose their race	10/10
users should be able to choose their appearance	10/10
users should be able to choose their background	10/10
users should be able to choose their ability scores	10/10
users should be able to choose their class	10/10
users should be able to choose all subsequent choices within those listed above	9/10, feats for human variant and attacks left out
users should be able to choose created characters to view	10/10
users should be able to increase their characters experience points	0/10, agreed didn't need to be implemented
users should be able to edit and delete created characters	4/10, can't be done easily, alter existing characters or delete them, but characters can be overwritten
users should be able to use their mouse to select options	10/10
users should be able to save their character sheets, to access whenever they need to	10/10

## Processing success

Processing requirements	Client success rating
The program should provide information about the choices the user makes	8/10, some information is given, but too much would be overwhelming
The program should recommend choices for classes	0/10, this isn't present
The program should be able to create characters and obtain all the appropriate information for choices	10/10
The program should allow Email and password recovery on login	0/10, this isn't here either, but Bram gives good reasons for it being left out
The program should store created characters for future access	10/10
The program should be able to generate random values when wanted e.g. ability scores	10/10
The program should assimilate all the information in one easy to use document, if possible, a character sheet.	10/10, this was particularly impressive
The program should calculate all Ability scores	10/10
The program should calculate all Ability modifiers	10/10
The program should calculate all Saving throws	10/10
The program should calculate all Skill modifiers	10/10
The program should calculate Armour class	10/10
The program should calculate Initiative	10/10
The program should calculate Hit points	10/10
The program should calculate Hit dice	10/10
The program should calculate Passive perception	10/10

## Design success

Design requirements	Client success rating
The layout should be minimalistic and easy to use	9/10, some of it could be laid out more clearly
there should be a login screen first then a main menu with the three main options presented	10/10
when creating characters the screen should bring up further choices	10/10
Left to right, top down familiar design	10/10
use of complementary colours	10/10
links to the main pages should be available on any part of the website in a bar at the top	10/10
Centralised design	10/10
Use of Pastel purple	10/10

## Output success

Output requirements	Client success rating
Output all the information to the correct boxes in the character sheet	10/10
The program should be able to output all 3 completed character sheets for the user to view	10/10
The program should be able to print it off character sheets with all the needed information	10/10
output all created characters to the GM so they can see what the players characters are	0/10, this, and GM accounts aren't present

## Final result:

**8.78/10**

## User testing results

### *Client's thoughts:*

From my use of the system, I have been able to make many characters, with everything being calculated correctly and creating a final character sheet that I can use with my club in game sessions. The final program far exceeded my expectations. Although there is room for improvement as we have discussed, such as being able to choose spells, and have attack values figured out as well as obviously levelling up characters. However I will be using the product in my sessions and it will be of great help to us.

## Final report

Clients thoughts on the final product

How well does this system meet your needs?

The Functionality is good and it covers the basis of character creation in a quick and easy to use format.

Could this system be used?

Yes, this can and will be used for our sessions.

Were the requirements met?

Yes

What problems are there with the system?

- lack of user feedback for mistakes e.g. a weight above 500 lbs
- missing functionality, the leveller, attacks, spells

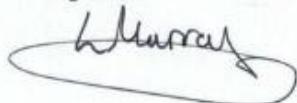
What aspects work particularly well?

- Maximum limit on checkboxes
- Simplistic design
- The 5 save slot system
- Printable character sheets

What improvements could be made?

- Signups should immediately log the user in
- alignment for centralised pages could be improved
- Fill the screen more
- 

Signature:



## Evaluation of user response

Mr. Murray's response to the product was both fair and full of praise. He agreed that the areas that were left out were reasonable and didn't limit the final product, but can and will be implemented at a later date.

The client mentioned the lack of spell choice, feat's, attacks and levelling. And he accepts it would take too long to implement for the reasons I will clarify fully now.

Spells have a lot of information about them. They have:

- casting time
- range
- components (what is needed to cast them)
- Duration
- Description of what they do
- Flavour text describing them
- What happens when they are cast at a higher level

Like this:

MAGIC MISSILE		
1st-level evocation		
<b>Casting Time:</b> 1 action		
<b>Range:</b> 120 feet		
<b>Components:</b> V, S		
<b>Duration:</b> Instantaneous		
You create three glowing darts of magical force. Each dart hits a creature of your choice that you can see within range. A dart deals $1d4 + 1$ force damage to its target. The darts all strike simultaneously, and you can direct them to hit one creature or several.		
<b>At Higher Levels.</b> When you cast this spell using a spell slot of 2nd level or higher, the spell creates one more dart for each slot level above 1st.		

In total the user gets to choose from 361 of these. It would be possible to implement this in the program, but it would take a huge amount of time.

THE FIGHTER		
Level	Proficiency Bonus	Features
1st	+2	Fighting Style, Second Wind
2nd	+2	Action Surge (one use)
3rd	+2	Martial Archetype
4th	+2	Ability Score Improvement
5th	+3	Extra Attack
6th	+3	Ability Score Improvement
7th	+3	Martial Archetype feature
8th	+3	Ability Score Improvement
9th	+4	Indomitable (one use)
10th	+4	Martial Archetype feature
11th	+4	Extra Attack (2)
12th	+4	Ability Score Improvement
13th	+5	Indomitable (two uses)
14th	+5	Ability Score Improvement
15th	+5	Martial Archetype feature
16th	+5	Ability Score Improvement
17th	+6	Action Surge (two uses), Indomitable (three uses)
18th	+6	Martial Archetype feature
19th	+6	Ability Score Improvement
20th	+6	Extra Attack (3)

Levelling would be a relatively simple process, each class would have its own page, where the user could enter how much experience their character has. Their character object would be loaded and depending on what level they reach, up to 20 levels of choices would be made. For example a fighter shown on the left.

also hit points, hit dice and all other ability score dependent variables would need to be recalculated each time. Each class also has branches that alter the character differently at certain levels. When an ability score would raise above 20, the user gets to choose feats, a separete issue. Finally, there are these 20 stages of levelling for every single class. This would be well over 250 different processes that would need to be calculated.

Feats would be troublesome to implement. There are over 40 of these small modifiers, that would all need to have their effects stored, some wouldn't need to be shown in a meaningful way on the character sheet other than by their name in the features section, however many alter numerical values, such as increasing initiative, armour class or letting the user learn a new language. These would be time consuming to implement.

Attacks are compared to the other three missing features the easiest to implement but are still time consuming to get finished within the timeframe. Attack stats involves creating new proficiency Booleans for every weapon type, as well as storing what ability scores affect what weapon. Letting the user choose from their equipped weapons what they want to use and have shown on their character sheet, then calculate attack modifiers and the damage modifier.

As agreed with the user these features will be implemented at a later date, however they are not integral to playing the game and the system created allows for the quick easy character creation that the user wanted.

## Extensions

The system is functional and currently a good core with which to create characters for sessions easily and quickly. The character sheets work perfectly and have been used in actual game sessions very successfully. However there are a large number of ways, I have come up with and pointed out by users that the system could be improved.

These extensions include:

- the recommendation feature to help new players decide on what choices to make, making it toggleable, so that experienced users aren't hampered by not helpful information. This would involve having a switch case block on each page for previously made choices and outputting recommendation information to a div on the side of the screen. This would all be contained in an if statement with a boolean value set by the user for if they want recommendation information.
- The account types, so that GM's can view their party's character sheets outside of games. This would involve a user choosing whether they are a GM on account creation and then having created characters being able to choose who their GM is from an existing list. Then adding those users sheets to the GM's character viewer.
- The character leveller. This would be a huge undertaking, as every class has a very large number of choices when levelling, it is not an urgent extension as once the character is created levelling is a relatively brief process with the book, as all the information the user needs is held in one chapter, unlike character creation where all the processing is scattered round the entire book. This would involve the user inputting the XP they have gained from playing, the program referencing that player to what level that puts them up to and iterating through the choices of each level until the user is at the correct level. However it would also alter ability scores and features, requiring values to be recalculated in an increasingly complex manner. Every class would have 20 levels they can iterate through.
- Tying E-mail addresses to accounts, so that passwords can be recovered would be a helpful feature that can be implemented at a later date. This would involve finding out how to validate E-mail addresses with JSP, and adding E-mail addresses to the login kind in the datastore.
- Calculating character attacks. Every character has a great number of attacks available to them, which go in the attacks section in the center of the character sheet. The user agreed in very early discussions that this and choosing spells was not necessary as the values change very frequently as characters pick up new weapons. This would involve storing the users weapons in a similar way to armour, then having the user choose which of their weapons they want to use. Depending on the weapon either their strength modifier or dexterity modifier is used, proficiencies are added, extra features are accounted for. Finally weapons damage would be output and the bonus to damage is calculated from the relevant modifier and features.
- Spells. There are a huge number of spells available to characters, and in certain cases players will not fill their spell sheet with all their spells, for example clerics and druids who have too many spells to fit on their sheet at once. There are websites designed exclusively to allow a user to choose their spells and print them off, such as <http://donjon.bin.sh/5e/spells/> the user agreed before production that implementing spell choice would be a huge undertaking and would not be possible to reasonably achieve. However it would be very desirable to completely create a spellcaster with the application, beyond calculating spellcasting scores. This would involve storing over 200 spells, their level and their meta information in arrays for each class. Have a page for users to select the number they're allowed and put the spells into the spell sheet.

- The ability for new races, classes and backgrounds would be a particularly interesting feature to implement. With the current architecture for the website it would be very difficult to implement, but as an extension it could be particularly fun for a GM running a session to implement new features. This would be very complicated and involve adapting the main class to have a setter method that gets what values to alter from arrays, and having a setter method for that setter to tell it which array values to use.

## Maintenance

### Adaptive maintenance

Adaptive maintenance should be unnecessary, as the rules for the game are almost certainly not going to change and as new game systems are introduced, there will be players still wanting to use this system of rules for their role playing games. However should the environment change the software for the server will be kept up to date, which should prevent bugs with changing systems.

### Perfective maintenance

Perfective maintenance is very simple to implement, as the local version can be altered to fix bugs in the deployed project. When ready to deploy the file can be archived to be returned to in the future, then the adapted version can be deployed, still using old Datastore information. If new Datastore columns are introduced, old entries will still be useable, as the Datastore can dynamically add columns to entities. This will allow tweaks to the efficiency of code and the pages to be made.

### Corrective maintenance

Although the testing showed the program to be mostly bug free, should any bugs be discovered, fixing is a simple case of making the project work locally in a safe environment then deploying the fixed version. This system of deployment means the website shouldn't be taken down for maintenance, however if it is necessary, the deployed version can be shut down to fix major bugs. The first aim of corrective maintenance should be to solve the error where users can go back pages and infinitely increase their ability scores as this is the main risk to making characters not allowed within the rule set.

# User Documentation

## Getting to the website

A web browser is required, whsbrpg supports all modern browsers, however is optimised for use on Google chrome with a 1920 x 1080p monitor. Simply open the browser on your device and navigate to the following URL:

<http://whsbrpg.appspot.com/>

For example on a Windows 7 PC with Google chrome, you would open the application and enter the URL into the search bar.



## Creating an account

When at the website you are presented with the following page:

### Signup Details

Username:

Password:

[Create account](#)

In order to create an account, simply click the link named 'Create account' and you will be redirected to this page:

# create account

Username:

Password:

Re-enter Password:

Input the username you would like to use and enter the password you want to use into the password and re-enter password field. If the username is taken or there was a mistake with your password, you will be told.

## Logging in

To log into your account enter your chosen username and password into the two respective fields. Then either press the submit button or press the enter key. Like shown:

## Signup Details

Username:

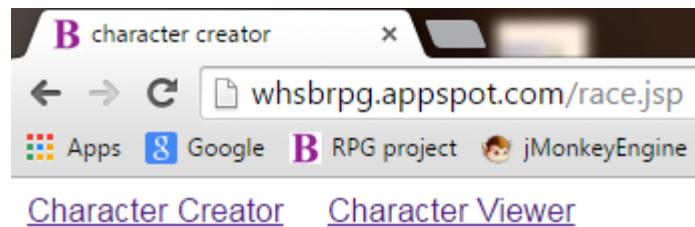
Password:

[Create account](#)

You will then be redirected to the first page of character creation.

## Creating a character

You can get to the start of character creation by either logging in or clicking the 'Character Creator' link in the navigation bar at the top left of every page as shown:



### Choosing your race

The first page of character creation is the race page. Here you choose what fantasy race you want your character to be:

## Bram's character creator

### Race

- High elf  Wood elf  Dark elf
- Hill dwarf  Mountain dwarf
- Lightfoot halfling  Stout halfling
  - Human  Human (variant)
  - Dragonborn
- Forest gnome  Rock gnome
  - Half-elf
  - Half-orc
  - Tiefling

### information:

You choose what race you want to be by clicking on the radio button to the left of the name you want. Then in the information section, some information about that race is shown to you.

As shown by the Tiefling option:

# Race

- High elf  Wood elf  Dark elf
- Hill dwarf  Mountain dwarf
- Lightfoot halfling  Stout halfling
- Human  Human (variant)
- Dragonborn
- Forest gnome  Rock gnome
- Half-elf
- Half-orc
- Tiefling

## information:

a half man half demon  
they get improved charisma, intelligence and resistance to fire

Some races then get other choices, which appear in the information area when you select them. For example if you choose to be a high elf, you get to learn a language of your choice:

# Race

- High elf  Wood elf  Dark elf
- Hill dwarf  Mountain dwarf
- Lightfoot halfling  Stout halfling
- Human  Human (variant)
- Dragonborn
- Forest gnome  Rock gnome
- Half-elf
- Half-orc
- Tiefling

## choose a language

Dwarvish  Giant  Gnomish  Goblin  Halfling  Orc  Abyssal  Celestial  Draconic  Deep Speech  Infernal  Primordial  Sylvan  Undercommon

## information:

High elves are the most magically talented of the elves  
they learn an additional language and begin with a magical cantrip. They have improved dexterity and intelligence.

If a race gives you additional choices like this you must pick one of the options shown or you will be unable to continue to the next stage of character creation. If checkboxes are created you will be told how many choices you need to make. You must make all the choices you are told to, so for half-elf when you are told to choose two ability scores to improve you have to check two of the boxes:

## choose your 2 improved ability scores

Strength  Dexterity  Constitution  Intelligence  Wisdom

When you have made all the choices necessary either click the submit button or press the enter key to continue to the next stage.

### Choosing your name and appearance

You are now shown this page:

## Name and Appearance

<b>character's name:</b> <input type="text"/>	<b>your name:</b> <input type="text"/>
Age: <input type="text"/>	Height <input type="text"/> foot <input type="text"/> inches
Eyes: <input type="text"/>	<input type="checkbox"/> random height and weight
Skin: <input type="text"/>	Weight <input type="text"/> pounds
Hair: <input type="text"/>	

Simply enter your choices into the textboxes, however for 'Age', 'Height' and 'Weight', you must enter numbers, any letters or punctuation will make you unable to continue. When entering your height, do not enter a number 12 or higher into the inches box.

If you want a random height and weight, simply check the box left of the words 'random height and weight'. If you do this any height or weight you entered will be ignored. When you have filled out all your choices press the submit button to continue.

### Choosing your background

You then get choose your characters background from the options shown on the page:

## Background

- Acolyte
- Charlatan
- Criminal  Spy
- Entertainer  Gladiator
- Folk Hero
- Guild Arisan  Guild Merchant
- Hermit
- Noble  Knight
- Outlander
- Sage
- Sailor  Pirate
- Soldier
- Urchin

**proficiencies:**

**languages:**

current languages: Common, Elf, Dwarvish

**equipment:**

**feature:**

This page works in the same way as the race page, you can chose a background and when you do it brings up relevant information and further choices in the area below. For example if you select the Acolyte background you get to choose two further languages:

## Background

- Acolyte
- Charlatan
- Criminal  Spy
- Entertainer  Gladiator
- Folk Hero
- Guild Arisan  Guild Merchant
- Hermit
- Noble  Knight
- Outlander
- Sage
- Sailor  Pirate
- Soldier
- Urchin

**proficiencies:**

You become proficient in religion and insight

**languages:**

current languages: Common, Elf, Dwarvish

You learn two new languages

Dwarvish  Elvish  Giant  Gnomish  Goblin  Halfling  Orc  Abyssal  Celestial  Draconic  Deep Speech  Infernal  Primordial  Sylvan  Undercommon

**equipment:**

You receive: A holy symbol, a prayer book, 5 sticks of incense, vestments, common clothes, 15 gold pieces

**feature:**

Shelter the faithful

When you have made all the choices you are told to, click the submit button or press enter to be sent to the next page.

## Choosing your alignment and personality

Now you will be able to select your characters alignment from a group of radio buttons. You will then be able to choose your characters personality trait, ideal, bond and flaw from four dropdown boxes. Simply click on a dropdown box and you are shown all the available options, choose the option you want the most for each box.

For example the acolyte page looks like this:

## Background

- Chaotic good    Good    Lawful good
- Chaotic neutral    Neutral    Lawful neutral
- Chaotic evil    Evil    Lawful evil

## Acolyte

### Personality trait

I idolize a particular hero of my faith, and constantly refer to that person's deeds and example. ▾

### Ideal

Tradition. The ancient traditions of worship and sacrifice must be preserved and upheld ▾

### Bond

I would die to recover an ancient relic of my faith that was lost long ago. ▾

### Flaw

I judge others harshly, and myself even more severley ▾

**Submit**

When you have made all your choices click the submit button or press the enter key.

## Choosing your characters back story

You will now be on a page with a large textbox. You have 500 characters to put into the back story behind your character, be creative and create whatever history you want. For example this is Mr. Example's back story

# Backstory

I grew up an orphan in a dark-elf city,  
raised to pray to the spider goddess  
Lolth. I was raised in the temple and  
did holy work for the priestess'.  
I learned how to fight in the  
underground city and began adventuring  
in my late teens.|

When you have settled on the back story press the submit button or press the enter key to continue.

## Choosing your ability scores

The next page gives you a choice between the three methods of getting your ability scores. If you are unsure click the preset button to use the standard set of numbers for created characters.

## Ability Scores

Use numbers as though you had rolled dice and used those values

use the standard number

use custom numbers by spending points

Otherwise simply choose whichever method you want to use to get your ability scores, by clicking on the button you want.

## Choosing random ability scores

On this page you are presented with a grid of radio buttons on the left and 6 numbers on the right:

**Random**

ability scores:	your random points
Strength: <input type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 7 <input type="radio"/> 12 <input type="radio"/> 7 <input type="radio"/> 10	14
Dexterity: <input type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 7 <input type="radio"/> 12 <input type="radio"/> 7 <input type="radio"/> 10	15
Constitution: <input type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 7 <input type="radio"/> 12 <input type="radio"/> 7 <input type="radio"/> 10	7
Intelligence: <input type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 7 <input type="radio"/> 12 <input type="radio"/> 7 <input type="radio"/> 10	12
Wisdom: <input type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 7 <input type="radio"/> 12 <input type="radio"/> 7 <input type="radio"/> 10	7
Charisma: <input type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 7 <input type="radio"/> 12 <input type="radio"/> 7 <input type="radio"/> 10	10

On the right are the 6 random numbers you got. You get to allocate one of these numbers to each of your ability scores. On the left are 6 rows, one for each score. The first row is strength; simply choose the strength score you want to have. Bear in mind you can't re use numbers, so if you decide you want 15 strength, none of your other scores can have 15. You can only choose one value to go in each ability score as well.

For example if I wanted:

Strength: 15

Dexterity: 12

Constitution: 14

Intelligence: 7

Wisdom: 7

Charisma: 10

you would check the boxes like this:

**ability scores:**

Strength: <input checked="" type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 7 <input type="radio"/> 12 <input type="radio"/> 7 <input type="radio"/> 10
Dexterity: <input type="radio"/> 14 <input checked="" type="radio"/> 15 <input type="radio"/> 7 <input checked="" type="radio"/> 12 <input type="radio"/> 7 <input type="radio"/> 10
Constitution: <input checked="" type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 7 <input type="radio"/> 12 <input type="radio"/> 7 <input type="radio"/> 10
Intelligence: <input type="radio"/> 14 <input type="radio"/> 15 <input checked="" type="radio"/> 7 <input type="radio"/> 12 <input type="radio"/> 7 <input type="radio"/> 10
Wisdom: <input type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 7 <input type="radio"/> 12 <input checked="" type="radio"/> 7 <input type="radio"/> 10
Charisma: <input type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 7 <input type="radio"/> 12 <input type="radio"/> 7 <input checked="" type="radio"/> 10

When you have decided on the ability scores you want, simply presss the submit button or press the enter key to continue.

## Choosing preset ability scores

On this page you are presented with a grid and numbers in the same way as for the random ability scores, however you always have the values: 15, 14, 13, 12, 10, and 8.

### Preset

	ability scores:	your points
Strength:	<input type="radio"/> 15 <input type="radio"/> 14 <input type="radio"/> 13 <input type="radio"/> 12 <input type="radio"/> 10 <input type="radio"/> 8	15
Dexterity:	<input type="radio"/> 15 <input type="radio"/> 14 <input type="radio"/> 13 <input type="radio"/> 12 <input type="radio"/> 10 <input type="radio"/> 8	14
Constitution:	<input type="radio"/> 15 <input type="radio"/> 14 <input type="radio"/> 13 <input type="radio"/> 12 <input type="radio"/> 10 <input type="radio"/> 8	13
Intelligence:	<input type="radio"/> 15 <input type="radio"/> 14 <input type="radio"/> 13 <input type="radio"/> 12 <input type="radio"/> 10 <input type="radio"/> 8	12
Wisdom:	<input type="radio"/> 15 <input type="radio"/> 14 <input type="radio"/> 13 <input type="radio"/> 12 <input type="radio"/> 10 <input type="radio"/> 8	10
Charisma:	<input type="radio"/> 15 <input type="radio"/> 14 <input type="radio"/> 13 <input type="radio"/> 12 <input type="radio"/> 10 <input type="radio"/> 8	8

This page operates in the same way as the random page, but with preset values.

## Choosing point bought ability scores

This page is intended for advanced players who can't get their characters the way they want with the preset values. It looks like this:

### Point bought

	ability scores:	your points
Strength:	<input type="radio"/> 8 <input type="radio"/> 9 <input type="radio"/> 10 <input type="radio"/> 11 <input type="radio"/> 12 <input type="radio"/> 13 <input type="radio"/> 14 <input type="radio"/> 15	27
Dexterity:	<input type="radio"/> 8 <input type="radio"/> 9 <input type="radio"/> 10 <input type="radio"/> 11 <input type="radio"/> 12 <input type="radio"/> 13 <input type="radio"/> 14 <input type="radio"/> 15	Score Cost
Constitution:	<input type="radio"/> 8 <input type="radio"/> 9 <input type="radio"/> 10 <input type="radio"/> 11 <input type="radio"/> 12 <input type="radio"/> 13 <input type="radio"/> 14 <input type="radio"/> 15	8 0
Intelligence:	<input type="radio"/> 8 <input type="radio"/> 9 <input type="radio"/> 10 <input type="radio"/> 11 <input type="radio"/> 12 <input type="radio"/> 13 <input type="radio"/> 14 <input type="radio"/> 15	9 1
Wisdom:	<input type="radio"/> 8 <input type="radio"/> 9 <input type="radio"/> 10 <input type="radio"/> 11 <input type="radio"/> 12 <input type="radio"/> 13 <input type="radio"/> 14 <input type="radio"/> 15	10 2
Charisma:	<input type="radio"/> 8 <input type="radio"/> 9 <input type="radio"/> 10 <input type="radio"/> 11 <input type="radio"/> 12 <input type="radio"/> 13 <input type="radio"/> 14 <input type="radio"/> 15	11 3
		12 4
		13 5
		14 7
		15 9

This system allows the user to pick any value they want for their ability scores between 8 and 15. As the table on the right shows, every score has a cost, and above that is how many points the user can spend. In the grid of scores the user can select, a value has to be put into each row, and the user has to spend all of their points for the character to be valid. For example:

## Point bought

ability scores:						your points			
Strength:	<input type="radio"/> 8	<input type="radio"/> 9	<input type="radio"/> 10	<input type="radio"/> 11	<input type="radio"/> 12	<input type="radio"/> 13	<input type="radio"/> 14	<input checked="" type="radio"/> 15	0
Dexterity:	<input type="radio"/> 8	<input type="radio"/> 9	<input type="radio"/> 10	<input type="radio"/> 11	<input type="radio"/> 12	<input checked="" type="radio"/> 13	<input type="radio"/> 14	<input type="radio"/> 15	
Constitution:	<input type="radio"/> 8	<input type="radio"/> 9	<input type="radio"/> 10	<input type="radio"/> 11	<input type="radio"/> 12	<input checked="" type="radio"/> 13	<input type="radio"/> 14	<input type="radio"/> 15	
Intelligence:	<input type="radio"/> 8	<input type="radio"/> 9	<input checked="" type="radio"/> 10	<input type="radio"/> 11	<input type="radio"/> 12	<input type="radio"/> 13	<input type="radio"/> 14	<input type="radio"/> 15	
Wisdom:	<input type="radio"/> 8	<input type="radio"/> 9	<input checked="" type="radio"/> 10	<input type="radio"/> 11	<input type="radio"/> 12	<input type="radio"/> 13	<input type="radio"/> 14	<input type="radio"/> 15	
Charisma:	<input type="radio"/> 8	<input type="radio"/> 9	<input type="radio"/> 10	<input type="radio"/> 11	<input checked="" type="radio"/> 12	<input type="radio"/> 13	<input type="radio"/> 14	<input type="radio"/> 15	
									Score Cost
									8 0
									9 1
									10 2
									11 3
									12 4
									13 5
									14 7
									15 9

When you have chosen a score for every row and the 'your points' section reads 0, you can click the submit button or press the enter key to continue onto the next section.

## Choosing your class

After ability score selection you are presented with this page:

## Choose a class

Barbarian  Bard  Cleric  Druid  Fighter  Monk  Paladin  Ranger  Rogue  Sorceror  Warlock  Wizard

## class information:

This page operates in the same way as race and background selection, however selecting a class will only bring up relevant information, there are no further choices on this page.

When you have decided on the class you want for example a fighter:

## Choose a class

Barbarian  Bard  Cleric  Druid  Fighter  Monk  Paladin  Ranger  Rogue  Sorceror  Warlock  Wizard

## class information:

**class:** Fighter

**Description:** A master of martial combat, skilled with a variety of weapons and armour

**Hit Die:** d10

**Primary Ability:** Strength or Dexterity

**Saving Throw Proficiencies:** Strength & Constitution

**Armour and Weapon Proficiencies:** All armour, shields, simple and martial weapons

You can then click the submit button or press the enter key to continue to the next page.

## Choosing class options

The last choice you make depends on your previous character choices. You are presented with a set or radio buttons and possibly checkboxes. Simply make a choice for every section, for what you want your character to have. For example on the fighter page:

**Bram's character creator**

**Fighter**

**Skills:**

choose two skills:  
 acrobatics  animal handling  athletics  history  insight  intimidation  perception  survival

**Equipment:**

choose a martial weapon  
 Chainmail or  Leather, longbow, and 20 arrows  
 choose a shield  
 Battleaxe  Flail  Glaive  Greataxe  Greatsword  Halberd  Lance  Longsword  Maul  Morningstar  Pike  Rapier  Scimitar  Shortsword  
 Trident  War pick  Warhammer  Whip  Blowgun  Crossbow, hand  Crossbow, heavy  Longbow  Net

and another martial weapon or a shield:  
 Shield  Battleaxe  Flail  Glaive  Greataxe  Greatsword  Halberd  Lance  Longsword  Maul  Morningstar  Pike  Rapier  Scimitar  Shortsword  
 Trident  War pick  Warhammer  Whip  Blowgun  Crossbow, hand  Crossbow, heavy  Longbow  Net

choose one of the following:  
 A light crossbow and 20 bolts or  two handaxes  
 choose a pack:  
 A dungeoneer's pack or  An explorers pack

choose a feature:  
 Archery: +2 to ranged attack rolls  
 Defence: +1 to AC  
 Dueling: +2 to one handed melee attacks (with no other weapons)  
 Great weapon fighting: reroll 1's and 2's for damage with two handed weapons  
 Protection: use your shield to give enemies in 5 ft disadvantage as a reaction  
 Two-weapon fighting: add your ability modifier to weapons used in your off hand

When you have made all the choices on the class options page, click the submit button or press the enter key to go to the final stage of character creation.

## Character confirmation

Finally you are shown a brief overview of your completed character that looks like this:

**Character**

Page 1

character name: M\_Example  
class: Fighter background: Acolyte real name: Bram  
race: Human gender: Male Lawful neutral xp: 0  
Strength: 15 mod: 2 save: 11 [AC: 15] [Initiative: 2] [Speed: 30]  
Dexterity: 15 mod: 2 save: 11  
Constitution: 13 mod: 1 save: 3 [HP: 11] [Hit Dice: 1D10]  
Intelligence: 11 mod: 0 save: 0  
Wisdom: 10 mod: 0 save: 0  
Charisma: 12 mod: 1 save: 1

personality traits: I quote(or misquote) sacred texts and proverbs in almost every situation  
ideals: Aspiration: I seek to prove myself worthy of my god's favor by matching my actions against his or her own teachings.  
bonds: I seek to preserve a sacred text that my enemies consider heretical and seek to destroy.  
flaws: Once I pick a goal, I become obsessed with it to the detriment of everything else in my life

false acrobatics: 2  
false animal handling: 0  
false arcane: 0  
true athletics: 4  
false deception: 1  
true history: 2  
true insight: 2  
false intimidation: 1  
false investigation: 0  
false medicine: 0  
false nature: 0  
true perception: 2  
false performance: 1  
true persuasion: 1  
true religion: 2  
false stealth: 0  
false survival: 2  
passive perception: 12

other proficiencies and languages:  
Common, Dwarven, Elvish, Orcish  
Longbows, Shortbows, Longwords, Shortwords, all armour, shields, simple weapons, martial weapons  
[gold: 15] equipment: A holy symbol, a prayer book, 5 sticks of incense, vestments, common clothes, leather, longbow, 20 arrows, Battleaxe, shield, two handaxes, explorers pack

features and traits: , Centrip, Shelter the Faithful, Archery, Second wind

page 2

character name: Bram  
age: 27 height: 4' 8" weight: 7 stone 0 pounds |  
eyes: Blue skin: Pale hair: Blonde |  
backstory: I grew up an orphan in a dark-elf city, raised to pray to the spider goddess Loth. I was raised in the temple and did holy work for the priestess. I learned how to fight in the underground city and began adventuring in my late teens.  
  
 save slot 1  save slot 2  save slot 3  save slot 4  save slot 5

If you see anything you want to alter, you can restart character creation in the navigation bar at the top left. If you are happy with your choices below page 2 on the right are 5 save slot radio buttons. Choose the save slot you want to use, then click the submit button or press the enter key to save your character. You will then be redirected to the character viewer.

## Viewing characters

You can get to the character viewer from any page after login by clicking the 'Character Viewer' navigation link in the top left, or by saving a character at the end of character creation. If you have not created a character yet, there is nothing of use to do on this section; it is recommended you click the link to character creation if this applies.

# Character Viewer

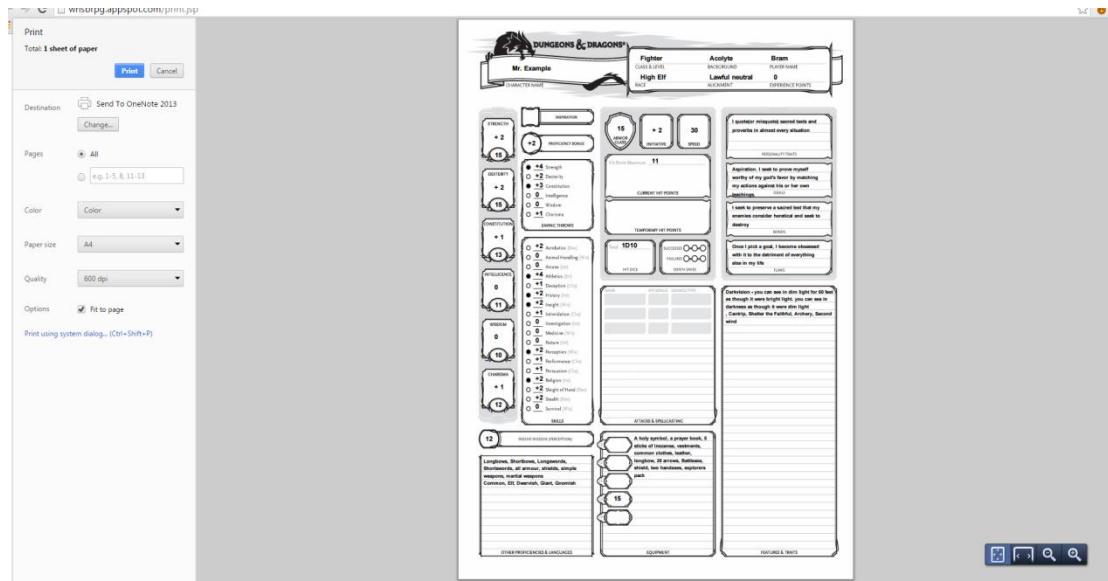
save slot 1  save slot 2  save slot 3  save slot 4  save slot 5  
 main sheet  background sheet  spell sheet

If you have saved a character already, click on the radio button for the save slot you saved it to, then select the sheet you want to view. Either the main sheet, the background sheet or if your character is a spell caster, the spell casting sheet. (if your character is not a spell caster, you will not be able to go on the spell casting page)

once you've chosen the sheet you want to view click the submit button or press the enter key.

## Character sheets

Each sheet has a button at the bottom the user can click to print off the sheet if they have a printer attached, however every page is better printed off by right clicking on the image and printing off that way, as shown.



Once each page has been printed off, you have finished and have a character sheet useable in a tabletop role playing game session.

## **Hardware and software requirements**

On a PC using IE7

Requirements
233MHz processor
Windows XP Service pack 2
Mouse or compatible pointing device
RAM (for the browser alone)
64MB for 32-bit windows XP/ Server 2003
128 MB for 64-bit windows XP/ Server 2003

The website will also work on OS X: 10.6, Android 4.0 or iOS 6.0 or later versions, when running Google Chrome.

## Troubleshooting

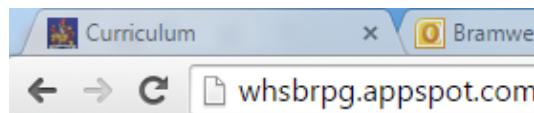
### I can't create an account

On the account page through this link <http://whsbrpg.appspot.com/account.jsp> you can create an account.

Reasons you may be unable to create an account include:

#### ***Your account name is already taken***

If this is the case, this error message should be visible:

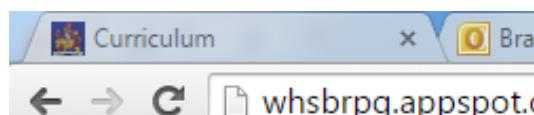


The username: spacegerbil is taken

Unfortunately if the username is taken you will have to select a different name.

#### ***The passwords you have entered are not identical***

If this is the case, this error message should be visible:

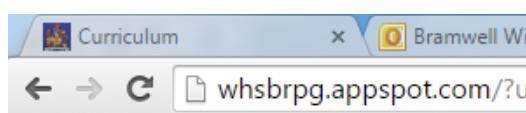


In this case you will need to re-enter your username and the two passwords, making sure that you have entered the same password both times.

### I can't login

On the login page through this link <http://whsbrpg.appspot.com/> you can login to your created account.

Reasons you will only be unable to login if you have entered the wrong username or password. For security reasons you are not informed which you have entered is incorrect. If this is the case you will see this message:



If this has happened you will need to carefully re-enter your username and password. In worst case scenarios a new account can be made by clicking the link to the create account page.

### **My page keeps refreshing and not sending me to the next page**

This can occur for two reasons:

### ***You have not filled out all the necessary fields on a page***

If you have not checked one of all the available radio buttons or filled out all the textboxes on a page, you will be unable to continue, as all choices are mandatory for a character to be created.

#### ***On the name and appearance selection page***

Exclusively on this page, you need to enter numbers instead of words in several columns:  
The age, feet, inches and pounds boxes:

Age:

Height  foot  inches

random height and weight

Weight  pounds

In each of these textboxes numbers such as 1, 4, 17 and 213 must be entered rather than words like apple, twelve or indigo.

## Final code

### CC Package

#### Coursework.java

```
package cc;
import java.io.Serializable;
/*
 * This class creates objects in which all of the player's choices are saved. It
 * also holds all the methods for processing numbers to then be saved to the
 * Datastore.
 */
public class coursework implements Serializable{/*Serializable prevents passing
errors later on */

    public String race;

    public String name;//real name
    public String pName;// player character name
    public String eyes;//colour
    public String skin;//colour
    public String hair;//colour
    public long age;
    public String size = "medium";

    public int height;
    public int weight;
    public int baseHeight;//set by race
    public int baseWeight;//set by race
    public int heightMod;//set by race
    public int weightMod;//set by race

    public String realH;//imperial units
    public String realW;//imperial units
    //background

    public String background;
    public String personality;//set by relevant jsp background page
    public String ideal;//set by relevant jsp background page
    public String bond;//set by relevant jsp background page
    public String flaw;//set by relevant jsp background page
    public String backstory;

    public long money;
    public String equipment= "";
    public String alignment;// set to a value from alignment array below
    public final String[] alignmentTable = { "Chaotic good", "Good", "Lawful
good",
                                             "Chaotic neutral", "Neutral", "Lawful neutral", "Chaotic
evil",
                                             "Evil", "Lawful evil" };//contains presets for player
alignment

    //abilities
    public long Str = 0; //all values that are saved to the datastore as ints,
are longs as the datastore only uses longs
    public long Dex = 0;
```

```

public long Con = 0;
public long Int = 0;
public long Wis = 0;
public long Cha = 0;

public long StrMod;//modifier, calculated in processing page
public long DexMod;
public long ConMod;
public long IntMod;
public long WisMod;
public long ChaMod;

public boolean strSave = false;//proficiency in ability save depends on
class
public boolean dexSave = false;
public boolean conSave = false;
public boolean intSave = false;
public boolean wisSave = false;
public boolean chaSave = false;

public long strSaveV = 0;//save value determined by proficiency and ability
modifier
public long dexSaveV = 0;
public long conSaveV = 0;
public long intSaveV = 0;
public long wisSaveV = 0;
public long chaSaveV = 0;

//proficiencies
public boolean athletics = false;//proficiency is controlled by either
background race or class.
public boolean acrobatics = false;
public boolean sleightOfHand = false;
public boolean stealth = false;
public boolean arcana = false;
public boolean history = false;
public boolean investigation = false;
public boolean nature = false;
public boolean religion = false;
public boolean animalHandling = false;
public boolean insight = false;
public boolean medicine = false;
public boolean perception = false;
public boolean survival = false;
public boolean deception = false;
public boolean intimidation = false;
public boolean performance = false;
public boolean persuasion = false;

public boolean doubleAthletics = false;//some features double proficiency
bonus. e.g. rogue expertise.
public boolean doubleAcrobatics = false;//These values if true double the
bonus for that skill
public boolean doubleSleight = false;
public boolean doubleStealth = false;
public boolean doubleArcana = false;
public boolean doubleHistory = false;

```

```

public boolean doubleInvestigation = false;
public boolean doubleNature = false;
public boolean doubleReligion = false;
public boolean doubleAnimal = false;
public boolean doubleInsight= false;
public boolean doubleMedicine = false;
public boolean doublePerception = false;
public boolean doubleSurvival = false;
public boolean doubleDeception = false;
public boolean doubleIntimidation = false;
public boolean doublePerformance = false;
public boolean doublePersuasion = false;

//values of the skills
public long athleticsV = 0;//value after proficiency, doubling and ther
modifier is calculated
public long acrobaticsV = 0;
public long sleightOfHandV = 0;
public long stealthV = 0;
public long arcanaV = 0;
public long historyV = 0;
public long investigationV = 0;
public long natureV = 0;
public long religionV = 0;
public long animalHandlingV = 0;
public long insightV = 0;
public long medicineV = 0;
public long perceptionV = 0;
public long survivalV = 0;
public long deceptionV = 0;
public long intimidationV = 0;
public long performanceV = 0;
public long persuasionV = 0;

public String otherProficiencies= "";//hold what other objects the
character adds their proficiency bonus to actions with
public long speed;//distance in feet the player can move a turn, depends on
class and race

public String vision = "";//what conditions the race can see in
public String languages= "";//languages the character can speak

public String features;//other bonuses the character can speak
public boolean defence = false;//if traited gain +1 ac

public String feats = "";//feats, obtainable by human variant and at higher
levels

//class
public String Class;
public int level = 1;//1 unless levelled up
public long xp = 0;//alters level at certain thresholds when implemented
public int profBonus = 2;//increases at higher levels to a maximum of 6
public String hitDice;//controlled by class, the dice used for healing and
number of dice.
public long hitPoints;//how much damage the character can take before dying

```

```

public int armour = 0;//int for each armour, alters ac
public boolean shield = false;//if the character has a shield they gain +2
AC
//spellcasting
public String spellAbil = "";//the ability the user uses for spells
public long spellDC = 111;//if set to 111, the character is not a
pellcaster. Determines how hard a spell is to avoid
public long spellMod = 111;//bonus to spells cast
public String spells = "";//list of spells learnt by a character

//final values
public boolean stealthDis = false;//given by heavy armour, gives
disadvantage to stealth rolls
public long ac;//determines how hard the character is to hit. Based on
armour and dexterity
public long initiative;//the bonus to rolls to go first in combat, based on
dexterity
public long passivePerception;//10 + perception
public coursework() {
    //constructor for instances of the class
}

/*
 * The races
 */
//the setter methods alter variables of this instance of the class that the
user interacts with.
public void HillDwarf(String toolProf) {//hill dwarf setter method with
parameter for choosing an extra proficiency
    this.race = "Hill Dwarf";//saves the race name
    this.Con = Con + 2;// constitution + 2
    this.speed = 25; // race speed = 25
    this.vision = "Darkvision - you can see in dim light for 60 feet as
though it were bright light."
        + " you can see in darkness as though it were dim
light";//saves the characters vision type
    this.features = "Dwarven resillience, Stonecunning";// saves the
racial features
    this.otherProficiencies = "Battleaxe, Handaxe, Throwing hammer,
Warhammer, Dwarven Toughness,"
        + toolProf;//proficiency in a tool given by race jsp
page
    this.languages = "Common, Dwarf";//saves the known racial languages
this.Wis = this.Wis + 1;// +1 wisdom

    this.baseHeight = 44; //saves the base height for random height
generation
    this.heightMod = processing.twoD4();//saves a number from 2 to 8 for
random height generation
    this.baseWeight = 115;//base height for random weight generation
    this.weightMod = processing.twoD6();// a random number from 2 to 12
for random weight generation
}
//for more information see the Hill Dwarf setter commented fully on line 169
public void MountainDwarf(String toolProf) {
    this.race = "Mountain Dwarf";
    this.Con = this.Con + 2;// constitution + 2
    this.speed = 25; // race speed = 25

```

```

        this.vision = "Darkvision - you can see in dim light for 60 feet as
though it were bright light."
                + " you can see in darkness as though it were dim
light";
        this.features = "Dwarven resillience, Stonecunning";
        this.otherProficiencies = "Battleaxe, Handaxe, Throwing hammer,
Warhammer, light armour, medium armour, "
                + toolProf;//proficiency in a tool given by race jsp
page
        this.languages = "Common, Dwarf";
        this.Str = this.Str + 2;// +2 strength

        this.baseHeight = 48;
        this.heightMod = processing.twoD4();
        this.baseWeight = 130;
        this.weightMod = processing.twoD6();
    }
//for more information see the Hill Dwarf setter commented fully on line 169
    public void HighElf(String languageIn) {
        this.race = "High Elf";
        this.speed = 30; // race speed = 30
        this.Dex = this.Dex + 2;// dexterity + 2
        this.vision = "Darkvision - you can see in dim light for 60 feet as
though it were bright light."
                + " you can see in darkness as though it were dim
light";
        this.perception = true;//proficiency in the perception skill
        this.languages = "Common, Elf, "+languageIn;// language given in
jsp page
        this.Int = this.Int + 1;// intelligence + 1
        this.otherProficiencies = "Longbows, Shortbows, Longswords,
Shortswords";
        this.features = this.features + ", Cantrip";
        // choose an extra cantrip

        this.baseHeight = 54;
        this.heightMod = processing.twoD10();
        this.baseWeight = 90;
        this.weightMod = processing.oneD4();
    }
//for more information see the Hill Dwarf setter commented fully on line 169
    public void WoodElf() {
        this.race = "Wood Elf";
        this.speed = 30; // race speed = 30
        this.Dex = this.Dex + 2;// dexterity + 2
        this.vision = "Darkvision - you can see in dim light for 60 feet as
though it were bright light."
                + " you can see in darkness as though it were dim
light";
        this.perception = true;//proficiency
        this.languages = "Common, Elf";
        this.features = "Trance, Fey Ancestry";// put in what they do
        this.Wis = this.Wis + 1;// wisdom + 1
        this.otherProficiencies = "Longbows, Shortbows, Longswords,
Shortswords";
        this.speed = 35; // race speed = 35
        this.features = this.features + ", Mask of the Wild";
    }

```

```

        this.baseHeight = 54;
        this.heightMod = processing.twoD10();
        this.baseWeight = 100;
        this.weightMod = processing.oneD4();
    }
//for more information see the Hill Dwarf setter commented fully on line 169
public void DarkElf() {
    this.race = "Dark Elf";
    this.speed = 30; // race speed = 30
    this.Dex = this.Dex + 2;// dexterity + 2
    this.perception = true;//proficiency in perception
    this.languages = "Common, Elf";
    this.features = "Trance, Fey Ancestry";// put in what they do
    this.Cha = this.Cha + 1;// +1 charisma
    this.vision = "Superior Darkvision - your Darkvision has a radius of
120 feet";
    this.otherProficiencies = "Rapiers, Shortwords, hand crossbows";
    this.features = this.features + ", Sunlight Sensitivity, Drow
Magic";

    this.baseHeight = 53;
    this.heightMod = processing.twoD6();
    this.baseWeight = 75;
    this.weightMod = processing.oneD6();
}
//for more information see the Hill Dwarf setter commented fully on line 169
public void LightfootHalfling() {
    this.race = "Lightfoot Halfling";
    this.Dex = this.Dex + 2;// +2 dexterity
    this.speed = 25;
    this.size = "small";//very few races are small, the standard size
for a race is medium, only the small races need setting
    this.languages = "Common, Halfling";
    this.features = "Lucky, Brave, Halfling Nimbleness";
    this.Cha = this.Cha + 1;// +1 charisma
    this.features = this.features + ", Naturally Stealthy";

    this.baseHeight = 31;
    this.heightMod = processing.twoD4();
    this.baseWeight = 35;
    this.weightMod = 1;
}
//for more information see the Hill Dwarf setter commented fully on line 169

public void StoutHalfling() {
    this.race = "Stout Halfling";
    this.Dex = this.Dex + 2;// +2 dexterity
    this.speed = 25;
    this.size = "small";
    this.languages = "Common, Halfling";
    this.features = "Lucky, Brave, Halfling Nimbleness";
    this.Con = this.Con + 1;// +1 constitution
    this.features = this.features + ", Stout Resilience";

    this.baseHeight = 31;
    this.heightMod = processing.twoD4();
    this.baseWeight = 35;
    this.weightMod = 1;
}

```

```

        }
//for more information see the Hill Dwarf setter commented fully on line 169

    public void Human(String language) { // standard human
        this.Str = this.Str + 1; //plus one to all ability scores
        this.Dex = this.Dex + 1;
        this.Con = this.Con + 1;
        this.Int = this.Int + 1;
        this.Wis = this.Wis + 1;
        this.Cha = this.Cha + 1;

        this.race = "human";
        this.speed = 30;
        this.languages = "Common, " + language; //bonus language of choice

        this.baseHeight = 56;
        this.heightMod = processing.twoD10();
        this.baseWeight = 115;
        this.weightMod = processing.twoD4();
    }
//for more information see the Hill Dwarf setter commented fully on line 169

    public void HumanVar(String language, int abil1, int abil2, int prof1,
                         String Feat) { // parameters for all the choices
        this.speed = 30;
        this.languages = "Common, " + language;
        switch (abil1) { //choose two abilities in race.jsp, they increase by
1
            case 1:
                this.Str = this.Str + 1;
                break;
            case 2:
                this.Dex = this.Dex + 1;
                break;
            case 3:
                this.Con = this.Con + 1;
                break;
            case 4:
                this.Int = this.Int + 1;
                break;
            case 5:
                this.Wis = this.Wis + 1;
                break;
            case 6:
                this.Cha = this.Cha + 1;
                break;
        }
        switch (abil2) {
            case 1:
                this.Str = this.Str + 1;
                break;
            case 2:
                this.Dex = this.Dex + 1;
                break;
            case 3:
                this.Con = this.Con + 1;
                break;
            case 4:

```

```

        this.Int = this.Int + 1;
        break;
    case 5:
        this.Wis = this.Wis + 1;
        break;
    case 6:
        this.Cha = this.Cha + 1;
        break;
    }
    chooseProf(prof1); //call function to choose one of all the skills
and become proficient in it
    this.race = "human";
    this.feats = Feat;

    this.baseHeight = 56;
    this.heightMod = processing.twoD10();
    this.baseWeight = 115;
    this.weightMod = processing.twoD4();
}
//for more information see the Hill Dwarf setter commented fully on line 169

public void Dragonborn(String ancestry) {
    this.race = "Dragonborn";
    this.speed = 30;
    this.Str = this.Str + 2; // str + 2
    this.Cha = this.Cha + 1; // +1 cha
    this.features = "Draconic Ancestry (" + ancestry
                    + "), Breath Weapon, Damage Resistance"; //choose a
colour of dragon
    this.languages = "Common, Draconic";

    this.baseHeight = 66;
    this.heightMod = processing.twoD8();
    this.baseWeight = 175;
    this.weightMod = processing.twoD6();
}
//for more information see the Hill Dwarf setter commented fully on line 169

public void ForestGnome() {
    this.race = "Forest Gnome";
    this.speed = 25;
    this.size = "small";
    this.Int = this.Int + 2; // int + 2
    this.Dex = this.Dex + 1; // +1 dex
    this.vision = "Darkvision - you can see in dim light for 60 feet as
though it were bright light."
                    + " you can see in darkness as though it were dim
light";
    this.features = "Gnome Cunning, Natural Illusionist, Speak with
Small Beasts ";
    this.languages = "Common, Gnomish";

    this.baseHeight = 35;
    this.heightMod = processing.twoD4();
    this.baseWeight = 35;
    this.weightMod = 1;
}
//for more information see the Hill Dwarf setter commented fully on line 169

```

```

public void RockGnome() {
    this.race = "Forest Gnome";
    this.speed = 25;
    this.size = "small";
    this.Int = this.Int + 2; // int + 2
    this.Con = this.Con + 1; // +1 con
    this.vision = "Darkvision - you can see in dim light for 60 feet as
though it were bright light."
        + " you can see in darkness as though it were dim
light";
    this.features = "Gnome Cunning, Artificer's Lore, Tinker";
    this.otherProficiencies = "Artisans tools";
    this.languages = "Common, Gnomish";

    this.baseHeight = 35;
    this.heightMod = processing.twoD4();
    this.baseWeight = 35;
    this.weightMod = 1;
}
//for more information see the Hill Dwarf setter commented fully on line 169

public void HalfElf(int abil1, int abil2, int prof1, int prof2,
                     String language) {
    this.race = "Half elf";
    this.speed = 30;
    this.Cha = this.Cha + 2; // + 2 cha
    switch (abil1) { // +1 to two skills
        case 1:
            this.Str = this.Str + 1;
            break;
        case 2:
            this.Dex = this.Dex + 1;
            break;
        case 3:
            this.Con = this.Con + 1;
            break;
        case 4:
            this.Int = this.Int + 1;
            break;
        case 5:
            this.Wis = this.Wis + 1;
            break;
        case 6:
            this.Cha = this.Cha + 1;
            break;
    }
    switch (abil2) {
        case 1:
            this.Str = this.Str + 1;
            break;
        case 2:
            this.Dex = this.Dex + 1;
            break;
        case 3:
            this.Con = this.Con + 1;
            break;
        case 4:
    }
}

```

```

        this.Int = this.Int + 1;
        break;
    case 5:
        this.Wis = this.Wis + 1;
        break;
    case 6:
        this.Cha = this.Cha + 1;
        break;
    }
    this.vision = "Darkvision - you can see in dim light for 60 feet as
though it were bright light."
        + " you can see in darkness as though it were dim
light";
    this.features = "Fey Ancestry, Skill Versatility";
    chooseProf(prof1); //gain proficiency in two skills
    chooseProf(prof2);
    this.languages = "Common, Elf, " + language; //gain a bonus language

    this.baseHeight = 57;
    this.heightMod = processing.twoD8();
    this.baseWeight = 110;
    this.weightMod = processing.twoD4();
}
//for more information see the Hill Dwarf setter commented fully on line 169

public void HalfOrc() {
    this.race = "Half orc";
    this.speed = 30;
    this.Str = this.Str + 2; // + 2 str
    this.Con = this.Con + 1; // +1 con
    this.vision = "Darkvision - you can see in dim light for 60 feet as
though it were bright light."
        + " you can see in darkness as though it were dim
light";
    this.features = "Menacing, Relentless Endurance, Savage Attacks";
    this.intimidation = true; //proficiency in intimidation
    this.languages = "Common, Orc";

    this.baseHeight = 58;
    this.heightMod = processing.twoD10();
    this.baseWeight = 140;
    this.weightMod = processing.twoD6();
}
//for more information see the Hill Dwarf setter commented fully on line 169

public void Tiefling() {
    this.race = "Tiefling";
    this.speed = 30;
    this.Cha = this.Cha + 2; // + 2 cha
    this.Int = this.Int + 1; // +1 int
    this.vision = "Darkvision - you can see in dim light for 60 feet as
though it were bright light."
        + " you can see in darkness as though it were dim
light";
    this.features = "Hellish Resistance, Infernal Legacy";
    this.languages = "Common, Infernal";

    this.baseHeight = 57;

```

```

        this.heightMod = processing.twoD8();
        this.baseWeight = 110;
        this.weightMod = processing.twoD4();
    }

/*
 * Background page, all appearance choices
 */

public void Background(String nameIn, String pNameIn, String eyesIn, String
skinIn,
                      String hairIn, int ageIn, Boolean randHW, String heightIn, int
weightIn) {//parameters for all the important choices
    this.name = nameIn;//set values to those chosen on the
nameAppearance.jsp page
    this.pName = pNameIn;
    this.age = ageIn;
    this.eyes = eyesIn;
    this.skin = skinIn;
    this.hair = hairIn;
    if (randHW) {// if the user wants a random height and weight, give
it.
        int Mod = heightMod;// makes sure the modifier doesn't change
        this.height = baseHeight + Mod;
        this.weight = baseWeight + (Mod * weightMod);
        this.realH = (this.height /12) + " " + (this.height % 12) +
"'"//convert to an imperial unit string
        this.realW = (this.weight /14) + " stone " + (this.weight %
14) + " pounds";//convert to an imperial unit string
    } else {
        this.realH = heightIn;//accepts a height
        this.realW = (weightIn /14) + " stone " + (weightIn % 14) + " "
pounds";
        //converts a weight in pounds to stone and pounds
    }
}
//in each specific background page an alignment is chosen from a table
public void setAlignment(int index) {
    this.alignment = alignmentTable[index];//set alignment to a value
from the array
}

/*
 * The backgrounds
 */

public void Acolyte(String Language1, String Language2) {//acolyte
background setter method chose two languages
    this.background = "Acolyte";//set background name
    this.religion = true;//gain proficiencies
    this.insight = true; // setting them to true
    this.languages = this.languages + ", " + Language1 + ", " +
Language2;//gain chosen languages from parameters
    this.money = 15;//set money in gold
    this.equipment = "A holy symbol, a prayer book, 5 sticks of
incense,"
                           + " vestments, common clothes";//give standard
equipment for the background

```

```

        if(this.features.equals("")){
            this.features = "Shelter the Faithful";//add features with no
comma if appropriate
        }else{
            this.features = features + ", Shelter the Faithful";//add
comma to start of features
        }
        /* all if statements checking if the string are empty are present to
make commas grammatically correct
depending on the race some fields may have information and others
may not, so fields need to be checked if
they are empty
*/
    }
//see the fully commented background setter on line 539
public void Charlatan(String scamTools) {// chose scam tools
    this.background = "Charlatan";
    this.sleightOfHand = true;//proficiencies
    this.deception = true;
    if(this.otherProficiencies.equals("")){//no comma
        this.otherProficiencies = "Disguise kit, Forgery kit";
    }else{//comma
        this.otherProficiencies = this.otherProficiencies
            + ", Disguise kit, Forgery kit";
    }
    this.money = 15;
    this.equipment = "A set of fine clothes, a disguise kit, tools of
your con including "
        + "(" + scamTools + ")";//equipment with chosen scam
    if(this.features.equals("")){
        this.features = "false Identity";//no comma
    }else{
        this.features = features + ", false Identity";
    }//comma

}
//see the fully commented background setter on line 539

public void Criminal(String gamingSet, Boolean variant) {
    this.background = "Criminal";
    if (variant)
        this.background = "Spy";//the different names are available
for character flavour, no changes beyond name
    this.stealth = true;
    this.deception = true;
    if(this.otherProficiencies.equals("")){//no comma
        this.otherProficiencies = "thieves tools, "+ gamingSet;
    }else{//comma
        this.otherProficiencies = this.otherProficiencies + ", thieves
tools, "
            + gamingSet;
    }
    this.money = 15;
    this.equipment = "A crowbar, A set of dark common clothing, a hood";
    if(this.features.equals("")){//no comma
        this.features = "Criminal contract";
    }else{//comma
        this.features = features + ", Criminal contract";
    }
}

```

```

        }

    }

//see the fully commented background setter on line 539

public void Entertainer(String instrument, Boolean variant) {
    this.background = "Entertainer";
    if (variant)
        this.background = "Gladiator";//flavour name
    this.acrobatics = true;
    this.performance = true;
    if(this.otherProficiencies.equals("")){//no comma
        this.otherProficiencies = "Disguise kit, "+
instrument;//proficient with instrument
    }else{//comma
        this.otherProficiencies = this.otherProficiencies + ", Disguise kit,
"
            + instrument;
    }
    this.money = 15;
    this.equipment = instrument //given instrument
        + ", the favor of an admirer, a costume";
    if(this.features.equals("")){
        this.features = "By popular demand";
    }else{
        this.features = features + ", By popular demand";
    }
}
//see the fully commented background setter on line 539

public void FolkHero(String artTools) {
    this.background = "Folk Hero";
    this.animalHandling = true;
    this.survival = true;
    if(this.otherProficiencies.equals("")){
        this.otherProficiencies = "vehicles(Land) " + artTools;
    }else{
        this.otherProficiencies = this.otherProficiencies + ",
vehicles(Land) "
            + artTools;
    }
    this.money = 10;
    this.equipment = artTools
        + ", a shovel, an iron pot, a set of common clothes";
    if(this.features.equals("")){
        this.features = "Rustic hospitality";
    }else{
        this.features = features + ", Rustic hospitality";
    }
}
//see the fully commented background setter on line 539

public void GuildArtisan(String artTools, String Language, boolean variant)
{
    this.background = "Guild Arisan";
    if (variant)
        this.background = "Guild Merchant";//flavour name

```

```

        this.insight = true;
        this.persuasion = true;
        if(this.languages.equals("")){
            this.languages = Language;
        }else{
            this.languages = this.languages + ", " + Language;
        }
        if(this.otherProficiencies.equals("")){
            this.otherProficiencies = artTools;
        }else{
            this.otherProficiencies = this.otherProficiencies + ", " + artTools;
        }
        this.money = 15;
        this.equipment = artTools
            + ", a letter of introduction from your guild, a "
            + "set of travellers clothes";
        if(this.features.equals("")){
            this.features = "Guild Membeship";
        }else{
            this.features = features + ", Guild Membeship";
        }
    }
//see the fully commented background setter on line 539

public void Hermit(String Language) {
    this.background = "Hermit";
    this.religion = true;
    this.medicine = true;
    this.languages = this.languages + ", " + Language;//bonus language
    if(this.otherProficiencies.equals("")){
        this.otherProficiencies = "Herbalism kit";
    }else{
        this.otherProficiencies = this.otherProficiencies + ", Herbalism
kit";
    }
    this.money = 5;
    this.equipment = " A scroll case stuffed with notes, a winter
blanket, a set"
        + " of common clothes, a herbalism kit";
    if(this.features.equals("")){
        this.features = "Discovery";
    }else{
        this.features = features + ", Discovery";
    }
}
//see the fully commented background setter on line 539

public void Noble(String gameSet, String Language, boolean variant) {
    this.background = "Noble";
    if (variant)
        this.background = "Knight";//change to name also changes feat
    this.history = true;
    this.persuasion = true;
    this.languages = this.languages + ", " + Language;
    if(this.otherProficiencies.equals("")){
        this.otherProficiencies = gameSet;
    }else{

```

```

        this.otherProficiencies = this.otherProficiencies + ", " +
gameSet;
    }
    this.money = 25;
    this.equipment = "A set of fine clothes, a signet ring, a scroll of
pedigree";
    if (variant) {//change to feat
        if(this.features.equals("")){//no comma
            this.features = "retainers";//knight trait
        }else{//comma
            this.features = features + ", retainers";
        }
    } else {
        if(this.features.equals("")){//no comma
            this.features = "position of privellige";//noble trait
        }else{//comma
            this.features = features + ", position of privellige";
        }
    }
}
//see the fully commented background setter on line 539

public void Outlander(String instrument, String Language) {
    this.background = "Outlander";
    this.athletics = true;
    this.survival = true;
    this.languages = this.languages + ", " + Language;//bonus language
    if(this.otherProficiencies.equals("")){
        this.otherProficiencies = instrument;//instrument
    }else{
        this.otherProficiencies = this.otherProficiencies + ", " +
instrument;
    }
    this.money = 10;
    this.equipment = "A staff, a hunting trap, a trophy from an animal "
        + "you killed, a set of travellers clothes";
    if(this.features.equals("")){
        this.features = "Wanderer";
    }else{
        this.features = features + ", Wanderer";
    }
}
//see the fully commented background setter on line 539

public void Sage(String Language1, String Language2) {
    this.background = "Sage";
    this.arcana = true;
    this.history = true;
    this.languages = this.languages + ", " + Language1 + ", " +
Language2;
    this.money = 10;
    this.equipment = "A bottle of black ink, a quill, a small knife, a
letter "
        + "from from a dead colleague posing a question you
have not been able"
        + " to answer, a set of common clothes";
    if(this.features.equals("")){
        this.features = "Researcher";
    }
}

```

```

        }else{
            this.features = features + ", Researcher";
        }
    }
//see the fully commented background setter on line 539

public void Sailor(Boolean variant) {// item can be a predetermined lucky
charm or a random trinket
    if (variant) {
        this.background = "Pirate";//variant name
        if(this.features.equals("")){//no comma
            this.features = "Bad Reputation";//variant trait
        }else{//comma
            this.features = features + ", Bad Reputation";
        }
    } else {
        this.background = "Sailor";
        if(this.features.equals("")){//no comma
            this.features = "Ship's Passage";
        }else{//comma
            this.features = features + ", Ship's Passage";
        }
    }
    this.athletics = true;
    this.perception = true;
    if(this.otherProficiencies.equals("")){
        this.otherProficiencies = "Navigator's tools,
Vehicles(water)";
    }else{
        this.otherProficiencies = this.otherProficiencies
            + ", Navigator's tools, Vehicles(water)";
    }
    this.money = 10;
    this.equipment = "A belaying pin (club), 50 feet of silk rope, a
lucky charm"
            + ", a set of common clothes";
}
//see the fully commented background setter on line 539

public void Soldier(String gameSet) {
    this.background = "Soldier";
    this.athletics = true;
    this.intimidation = true;
    if(this.otherProficiencies.equals("")){
        this.otherProficiencies = gameSet
            + ", Vehicles(Land)";
    }else{
        this.otherProficiencies = this.otherProficiencies + ", " +
gameSet
            + ", Vehicles(Land)";
    }
    this.money = 10;
    this.equipment = "Insignia of rank, trophy from an enemy, " +
gameSet
            + ", common clothes";
    if(this.features.equals("")){
        this.features = "Military Rank";
    }
}

```

```

        }else{
            this.features = features + ", Military Rank";
        }
    }
//see the fully commented background setter on line 539

public void Urchin() {
    this.background = "Urchin";
    this.sleightOfHand = true;
    this.stealth = true;
    if(this.otherProficiencies.equals("")){
        this.otherProficiencies = "disguise kit, thieves tools";
    }else{
        this.otherProficiencies = this.otherProficiencies
                                + ", disguise kit, thieves tools";
    }
    this.money = 10;
    this.equipment = "A small knife, a map of the city you grew up in, a
pet mouse,"
                    + " a token to remember your parents by, a set of
common clothes";
    if(this.features.equals("")){
        this.features ="City Secrets";
    }else{
        this.features = features + ", City Secrets";
    }
}

/*
 * classes
 */
//chosen in class.jsp, then extras in the relevant class jsp page

public void barbaian(int skill1, int skill2, String item1, String item2){
//barbarian class setter with parameters for choices
    this.Class = "Barabarian";//set class name
    this.hitDice = level +"D12";//always equal to a number equal to
level of a dice determined by the class, here: 1 12 sided
    this.hitPoints = (12 + processing.calcModifier(Con));//hitpoints
depend on the base and players constitution modifier
    if(this.otherProficiencies.equals("")){//add other proficiencies
without a comma if appropriate
        this.otherProficiencies += "light armour, medium armour,
shields, simple weapons, martial weapons";
    }else{//add other proficiencies with a comma
        this.otherProficiencies += ", light armour, medium armour,
shields, simple weapons, martial weapons";
    }
    this.strSave = true;//all classes gain proficiency in two ability
saves
    this.conSave = true; //by setting them to true
    switch(skill1){//accepts parameters of chosen skills and sets the
appropriate Booleans to true
        case 1:
            this.animalHandling = true;

```

```

        break;
    case 2:
        this.athletics = true;
        break;
    case 3:
        this.intimidation = true;
        break;
    case 4:
        this.nature = true;
        break;
    case 5:
        this.perception = true;
        break;
    case 6:
        this.survival = true;
        break;
    }
    switch(skill2){ //accepts the second skill choice parameter
    case 1:
        this.animalHandling = true;
        break;
    case 2:
        this.athletics = true;
        break;
    case 3:
        this.intimidation = true;
        break;
    case 4:
        this.nature = true;
        break;
    case 5:
        this.perception = true;
        break;
    case 6:
        this.survival = true;
        break;
    }
    if(this.equipment.equals("")){//add chosen items to equipment with
checks for whether or not to add commas
        this.equipment += item1;
    }else{
        this.equipment += ", " + item1;
    }
    this.equipment += ", " + item2; //this will always have a comma
    this.equipment += ", an explorers pack, four javelins";//final
equipment appendment
    this.features += ", Rage, unarmoured defence";//unarmoured defence
alters ac
    this.armour = 31;//unnarmoured defence ac reference value
}
//for a completely commented character setter, see line 836
public void bard(String inst1, String inst2, String inst3, int skill1, int
skill2, int skill3, String item1, String item2, String item3){
    this.Class = "Bard";
    this.hitDice = level + "D8";//1 8 sided dice for healing
    this.hitPoints = 8 + processing.calcModifier(Con);//base for bard is
8
    if(this.otherProficiencies.equals("")){


```

```

        this.otherProficiencies += "light armour, simple weapons, hand
crossbows, longswords, rapiers, shortswords";
    }else{
        this.otherProficiencies += ", light armour, simple weapons,
hand crossbows, longswords, rapiers, shortswords";
    }
    this.dexSave = true;
    this.chaSave = true;
    this.otherProficiencies += ", " + inst1 + ", " + inst2 +", " +
inst3;// add proficiencies in chosen instruments
    this.chooseProf(skill11);//proficient with three skills
    this.chooseProf(skill12);
    this.chooseProf(skill13);
    this.spellAbil = "Charisma";//spellcasting ability modifier
    this.spellDC = 8 + this.profBonus +
processing.calcModifier(this.Cha);//all based on the relevant ability modifier
    this.spellMod = this.profBonus +
processing.calcModifier(this.Cha);//here the modifier is charisma
    this.features +=", Spellcasting, ritual casting, spellcasting
focus(instrument), Bardic inspiration(d6)";
    //four known 1st level spells, 2 spell slots, 2 cantrips
    if(this.equipment.equals("")){
        this.equipment += item1 + ", " + item2 + ", " + item3 +",
Leather armour, a dagger";
    }else{
        this.equipment += ", " + item1 + ", " + item2 + ", " + item3 +
", Leather armour, a dagger";
    }
    this.armour = 2;//leather armour reference
}
//for a completely commented character setter, see line 836
public void cleric(int skill11, int skill12, boolean item1, int item2, String
item3, boolean item4, int domain){
    this.Class = "Cleric";
    this.hitDice = level +"D8";
    this.hitPoints = 8 + processing.calcModifier(Con);
    if(this.otherProficiencies.equals("")){
        this.otherProficiencies += "light armour, medium armour,
shields, all simple weapons";
    }else{
        this.otherProficiencies += ", light armour, medium armour,
shields, all simple weapons";
    }
    this.wisSave = true;
    this.chaSave = true;
    switch(skill11){
    case 1 :
        this.history = true;
        break;
    case 2 :
        this.insight = true;
        break;
    case 3 :
        this.medicine = true;
        break;
    case 4 :
        this.persuasion = true;
        break;
    }
}

```

```

        case 5 :
            this.religion = true;
            break;
    }
    switch(skill2){
    case 1 :
        this.history = true;
        break;
    case 2 :
        this.insight = true;
        break;
    case 3 :
        this.medicine = true;
        break;
    case 4 :
        this.persuasion = true;
        break;
    case 5 :
        this.religion = true;
        break;
    }
    if(!this.equipment.equals("")){
        this.equipment += ", ";
    }
    if(item1){
        this.equipment += "mace";//set chosen items
    }else{
        this.equipment += "warhammer";
    }
    switch(item2){
    case 1 :
        this.equipment += ", scale mail";
        this.armour = 13;
        break;
    case 2 :
        this.equipment += ", leather armour";
        this.armour = 2;
        break;
    case 3 :
        this.equipment += ", chain mail";
        this.armour = 22;
        break;
    }
    if(item3.equals("")){
        this.equipment += ", a light crossbow, 20 bolts";
    }else{
        this.equipment += ", "+item3;
    }
    if(item4){
        this.equipment += ", priests pack";
    }else{
        this.equipment += ", explorers pack";
    }
    // 3 cantrips, all the spells, 2 level 1 spell slots
    this.spellAbil = "Wisdom";//spellcasting modifier is wisdom
    this.spellDC = 8 + this.profBonus +
processing.calcModifier(this.Wis);
    this.spellMod = this.profBonus + processing.calcModifier(this.Wis);

```

```

        this.features +=", ritual casting";
        this.features += ", divine domain: ";
        switch(domain){//different domains give different features and
spells
    case 1 :
        this.features += "Knowledge, Blessings of
knowledge";//blessings of knowledge method gives extra bonuses
        this.spells += "command, identify";
        break;
    case 2 :
        this.features += "Life";
        this.spells += "bless, cure wounds";
        this.otherProficiencies +=", heavy armour";
        this.features += ", Disciple of life";
        break;
    case 3 :
        this.features += "Light, Warding flame";
        this.spells += "bunring hands, faerie fire";
        //bonus cantrip light
        break;
    case 4 :
        this.features += "Nature, acolyte of nature";
        this.spells += "animal friendship, speak with
animals";//acolyte of nature method gives extra bonuses
        this.otherProficiencies +=", heavy armour";
        break;
    case 5 :
        this.features += "Tempest, wrath of the storm";
        this.spells += "fog cloud, thunderwave";
        this.otherProficiencies +=",martial weapons, heavy armour";
        break;
    case 6 :
        this.features += "Trickery, blessing of the trickster";
        this.spells += "charm person, disguise self";
        break;
    case 7 :
        this.features += "War, war priest";
        this.spells += "divine favor, shield of faith";
        this.otherProficiencies +=",martial weapons, heavy armour";
        break;
    }
}
public void blessKnowledge(String lang1, String lang2, int prof1, int
prof2){//for knowledge domain
    this.languages += ", " + lang1 + ", " + lang2;//gain two languages
    switch(prof1){
        case 1://gain proficiency in two skills and double their prof
bonus's
            this.arcana = true;
            this.doubleArcana = true;
            break;
        case 2:
            this.history = true;
            this.doubleHistory = true;
            break;
        case 3:
            this.nature = true;
    }
}

```

```

        this.doubleNature = true;
        break;
    case 4:
        this.religion = true;
        this.doubleReligion = true;
        break;
    }//double prof bonus!
switch(prof2){
case 1:
    this.arcana = true;
    this.doubleArcana = true;
    break;
case 2:
    this.history = true;
    this.doubleHistory = true;
    break;
case 3:
    this.nature = true;
    this.doubleNature = true;
    break;
case 4:
    this.religion = true;
    this.doubleReligion = true;
    break;
}
}

public void acolyteOfNature(int prof){// for nature domain
switch(prof){//proficiency in one of these skills
case 1:
    this.animalHandling = true;
    break;
case 2:
    this.nature = true;
    break;
case 3:
    this.survival = true;
    break;
}
//plus cantrip
}
//for a completely commented character setter, see line 836
public void druid(int skill1, int skill2, String item1, String item2){
    this.Class = "Druid";
    this.hitDice = level +"D8";
    this.hitPoints = 8 + processing.calcModifier(Con);
    if(!this.otherProficiencies.equals("")){
        this.otherProficiencies += ", ";
    }
    this.otherProficiencies += "light armour, medium armour, shields,
clubs, daggers, darts, javelins, maces,"
        + " quaterstaffs, scimitars, sickles, slings, spears,
herbalism kit";
    this.intSave = true;
    this.wisSave = true;
switch(skill1){
case 1:
    this.arcana = true;
    break;

```

```

        case 2:
            this.animalHandling = true;
            break;
        case 3:
            this.insight = true;
            break;
        case 4:
            this.medicine = true;
            break;
        case 5:
            this.nature = true;
            break;
        case 6:
            this.perception = true;
            break;
        case 7:
            this.religion = true;
            break;
        case 8:
            this.survival = true;
            break;
    }
    switch(skill2){
        case 1:
            this.arcana = true;
            break;
        case 2:
            this.animalHandling = true;
            break;
        case 3:
            this.insight = true;
            break;
        case 4:
            this.medicine = true;
            break;
        case 5:
            this.nature = true;
            break;
        case 6:
            this.perception = true;
            break;
        case 7:
            this.religion = true;
            break;
        case 8:
            this.survival = true;
            break;
    }
    if(!this.equipment.equals("")){
        this.equipment += ", ";
    }
    if(item1.equals ""){
        this.equipment += "a wooden shield";
        this.shield = true;
    }else{
        this.equipment += item1;
    }
    if(item2.equals ""){

```

```

        this.equipment += ", a scimitar";
    }else{
        this.equipment += ", " + item2;
    }
    this.equipment += ", leather armour, explorer's pack, druidic focus";
    this.armour = 2;
    this.spellAbil = "Wisdom";
    this.spellDC = 8 + this.profBonus + processing.calcModifier(this.Wis);
    this.spellMod = this.profBonus + processing.calcModifier(this.Wis);
    this.features +=", Spellcasting, ritual casting";
    this.languages += ", druidic";
    //all spells 2 cantrips, 2 spell slots
}
//for a completely commented character setter, see line 836
public void fighter(int skill1, int skill2, boolean item1, String item2,
String item3, boolean item4, boolean item5, int feature){
    this.Class = "Fighter";
    this.hitDice = level +"D10";
    this.hitPoints = 10 + processing.calcModifier(Con);
    if(!this.otherProficiencies.equals("")){
        this.otherProficiencies += ", ";
    }
    this.otherProficiencies += "all armour, shields, simple weapons,
martial weapons";
    this.strSave = true;
    this.conSave = true;
    switch(skill1){
        case 1:
            this.acrobatics = true;
            break;
        case 2:
            this.animalHandling = true;
            break;
        case 3:
            this.athletics = true;
            break;
        case 4:
            this.history = true;
            break;
        case 5:
            this.insight = true;
            break;
        case 6:
            this.intimidation = true;
            break;
        case 7:
            this.perception = true;
            break;
        case 8:
            this.survival = true;
            break;
    }
    switch(skill2){
        case 1:
            this.acrobatics = true;
            break;
        case 2:
            this.animalHandling = true;

```

```

        break;
    case 3:
        this.athletics = true;
        break;
    case 4:
        this.history = true;
        break;
    case 5:
        this.insight = true;
        break;
    case 6:
        this.intimidation = true;
        break;
    case 7:
        this.perception = true;
        break;
    case 8:
        this.survival = true;
        break;
    }
    if(!this.equipment.equals("")){
        this.equipment += ", ";
    }
    if(item1){
        this.equipment += "chainmail";
        this.armour = 22;//chainmail armour reference
    }else{
        this.equipment += "leather, longbow, 20 arrows";
        this.armour = 2;//leather armour reference
    }
    if(item3.equals("")){
        this.equipment += ", "+item2+", shield";
        this.shield = true; // make for with one handed
    }else{
        this.equipment += ", " + item2 + ", " + item3;
    }
    if(item4){
        this.equipment += ", light crossbow and 20 bolts";
    }else{
        this.equipment += ", two handaxes";
    }
    if(item5){
        this.equipment += ", dungeoneers pack";
    }else{
        this.equipment += ", explorers pack";
    }

    if(!this.features.equals("")){
        this.features += ", ";
    }
    switch(feature){
    case 1:
        this.features += "Archery";
        break;
    case 2:
        this.features += "Defense";
        this.defence = true;//+1 to ac
        break;
    }

```

```

        case 3:
            this.features += "Dueling";
            break;
        case 4:
            this.features += "Great weapon fighting";
            break;
        case 5:
            this.features += "Protection";
            break;
        case 6:
            this.features += "Two-weapon fighting";
            break;
    }
    this.features += ", Second wind";

}
//for a completely commented character setter, see line 836
public void monk(String i, int skill1, int skill2, String item1, String
item2){
    this.Class = "Monk";
    this.hitDice = level +"D8";
    this.hitPoints = 8 + processing.calcModifier(Con);
    if(!this.otherProficiencies.equals("")){
        this.otherProficiencies += ", ";
    }
    this.otherProficiencies += "simple weapons, shortswords, " + i;//add chosen item proficiency
    this.strSave = true;
    this.dexSave = true;
    switch(skill1){
        case 1:
            this.acrobatics = true;
            break;
        case 2:
            this.athletics = true;
            break;
        case 3:
            this.history = true;
            break;
        case 4:
            this.insight = true;
            break;
        case 5:
            this.religion = true;
            break;
        case 6:
            this.stealth = true;
            break;
    }
    switch(skill2){
        case 1:
            this.acrobatics = true;
            break;
        case 2:
            this.athletics = true;
            break;
        case 3:
            this.history = true;

```

```

        break;
    case 4:
        this.insight = true;
        break;
    case 5:
        this.religion = true;
        break;
    case 6:
        this.stealth = true;
        break;
    }
    if(!this.equipment.equals("")){
        this.equipment += ", ";
    }
    if(item1.equals("")){
        this.equipment += "shortsword";
    }else{
        this.equipment += item1;
    }
    if(item2.equals("1")){
        this.equipment += ", dungeoneer's pack";
    }else{
        this.equipment += ", explorer's pack";
    }
    this.equipment += ", 10 darts";
    this.features += ", unarmed defence, martial
arts(1d4)";//increase ac, unarmed attacks d4
    this.armour = 32;//unarmoured defence ac bonus
}
//for a completely commented character setter, see line 836
public void paladin(int skill1,int skill2, String item1, String item2,
String item3, boolean pack){
    this.Class = "Paladin";
    this.hitDice = level +"D10";
    this.hitPoints = 10 + processing.calcModifier(Con);
    if(!this.otherProficiencies.equals("")){
        this.otherProficiencies += ", ";
    }
    this.otherProficiencies += "all armour, shields, simple weapons,
martial weapons";
    this.wisSave = true;
    this.chaSave = true;
    switch(skill1){
    case 1:
        this.athletics = true;
        break;
    case 2:
        this.insight = true;
        break;
    case 3:
        this.intimidation = true;
        break;
    case 4:
        this.medicine = true;
        break;
    case 5:
        this.persuasion = true;
        break;
    }
}

```

```

        case 6:
            this.religion = true;
            break;
    }
    switch(skill2){
    case 1:
        this.athletics = true;
        break;
    case 2:
        this.insight = true;
        break;
    case 3:
        this.intimidation = true;
        break;
    case 4:
        this.medicine = true;
        break;
    case 5:
        this.persuasion = true;
        break;
    case 6:
        this.religion = true;
        break;
    }
    if(!this.equipment.equals("")){
        this.equipment += ", ";
    }
    if(item2.equals("")){
        this.equipment += item1 + ", shield";
        this.shield = true;
    }else{
        this.equipment += item1 + ", " + item2;
    }
    if(item3.equals("")){
        this.equipment += ", five javelins";
    }else{
        this.equipment += ", " + item3;
    }
    if(pack){
        this.equipment += ", priest's pack";
    }else{
        this.equipment += ", explorer's pack";
    }
    this.equipment += ", chain mail, holy symbol";
    this.armour = 22;
    this.features += ", Divine sense, lay on hands";//things you can do
}
//for a completely commented character setter, see line 836
public void ranger(int skill1, int skill2, boolean armour, String weapon1,
String weapon2, boolean pack, String favouredEnemy){
    this.Class = "Ranger";
    this.hitDice = level +"D10";
    this.hitPoints = 10 + processing.calcModifier(Con);
    if(!this.otherProficiencies.equals("")){
        this.otherProficiencies += ", ";
    }
    this.otherProficiencies += "light armour, medium armour, shields,
simple weapons, martial weapons";

```

```

this.strSave = true;
this.dexSave = true;
switch(skill1){
case 1:
    this.animalHandling = true;
    break;
case 2:
    this.athletics = true;
    break;
case 3:
    this.insight = true;
    break;
case 4:
    this.investigation = true;
    break;
case 5:
    this.nature = true;
    break;
case 6:
    this.perception = true;
    break;
case 7:
    this.stealth = true;
    break;
case 8:
    this.survival = true;
    break;
}
switch(skill2){
case 1:
    this.animalHandling = true;
    break;
case 2:
    this.athletics = true;
    break;
case 3:
    this.insight = true;
    break;
case 4:
    this.investigation = true;
    break;
case 5:
    this.nature = true;
    break;
case 6:
    this.perception = true;
    break;
case 7:
    this.stealth = true;
    break;
case 8:
    this.survival = true;
    break;
}
if(!this.equipment.equals("")){
    this.equipment += ", ";
}
if(armour){

```

```

        this.equipment += "scale mail";
        this.armour = 13;
    }else{
        this.equipment += "leather armour";
        this.armour = 2;
    }
    if(weapon1.equals("")){
        this.equipment += ", two shortswords";
    }else{
        this.equipment += ", " + weapon1 + ", " + weapon2;
    }
    if(pack){
        this.equipment += ", dungeoneer's pack";
    }else{
        this.equipment += ", explorer's pack";
    }
    this.equipment += ", longbow, a quiver of 20 arrows";
    this.features += ", Favored enemy("+favouredEnemy+"), Natural
explorer";//favoured enemy can either be a master type
    //or two humanoid races
}
//for a completely commented character setter, see line 836
public void rogue(int skill1, int skill2, int skill3, int skill4, String
item1, String item2, String item3){
    this.Class = "Rogue";
    this.hitDice = level +"D8";
    this.hitPoints = 8 + processing.calcModifier(Con);
    if(!this.otherProficiencies.equals ""){
        this.otherProficiencies += ", ";
    }
    this.otherProficiencies += "light armour, simple weapons, hand
crossbows,"
        + " longswords, rapiers, shortswords, thieves tools";
    this.dexSave = true;
    this.intSave = true;
    for(int x = 0; x < 4; x++){//gain proficiency in four skills
        int s = 0;
        switch(x){
            case 0:
                s = skill1;
                break;
            case 1:
                s = skill2;
                break;
            case 2:
                s = skill3;
                break;
            case 3:
                s = skill4;
                break;
        }
        switch(s){
            case 1:
                this.acrobatics = true;
                break;
            case 2:
                this.athletics = true;
                break;
        }
    }
}

```

```

        case 3:
            this.deception = true;
            break;
        case 4:
            this.insight = true;
            break;
        case 5:
            this.intimidation = true;
            break;
        case 6:
            this.investigation = true;
            break;
        case 7:
            this.perception = true;
            break;
        case 8:
            this.performance = true;
            break;
        case 9:
            this.persuasion = true;
            break;
        case 10:
            this.sleightOfHand = true;
            break;
        case 11:
            this.stealth = true;
            break;
    }
}
if(!this.equipment.equals("")){
    this.equipment += ", ";
}
this.equipment += item1 + item2 + item3 + ", leather armour, two
daggers, thieve's tools";
this.armour = 2;
this.features += ", Expertise, Sneak attack, Thieve's cant";
}

public void expertise(int skill1, int skill2){//double proficiency bonus in
two skills the player is proficient in
    int e = 0;
    for(int x = 0; x<2;x++){
        switch(x){
            case 0:
                e = skill1;
                break;
            case 1:
                e = skill2;
                break;
        }
    }
    switch(e){
        case 1:
            this.doubleAcrobatics = true;
            break;
        case 2:
            this.doubleAnimal = true;
            break;
        case 3:
    }
}

```

```

                this.doubleArcana = true;
                break;
            case 4:
                this.doubleAthletics = true;
                break;
            case 5:
                this.doubleDeception = true;
                break;
            case 6:
                this.doubleHistory = true;
                break;
            case 7:
                this.doubleInsight = true;
                break;
            case 8:
                this.doubleIntimidation = true;
                break;
            case 9:
                this.doubleInvestigation = true;
                break;
            case 10:
                this.doubleMedicine = true;
                break;
            case 11:
                this.doubleNature = true;
                break;
            case 12:
                this.doublePerception = true;
                break;
            case 13:
                this.doublePerformance = true;
                break;
            case 14:
                this.doublePersuasion = true;
                break;
            case 15:
                this.doubleReligion = true;
                break;
            case 16:
                this.doubleSleight = true;
                break;
            case 17:
                this.doubleStealth = true;
                break;
            case 18:
                this.doubleSurvival = true;
                break;
        }
    }
}

//for a completely commented character setter, see line 836
public void sorcerer(int skill1, int skill2, String item1, String item2,
String item3, int origin){
    this.Class = "Sorcerer";
    this.hitDice = level +"D6";
    this.hitPoints = 6 + processing.calcModifier(Con);
    if(!this.otherProficiencies.equals("")){
        this.otherProficiencies += ", ";
    }
}

```

```

        }
        this.otherProficiencies += "Daggers, darts, slings, quaterstaffs,
light crossbows";
        this.conSave = true;
        this.chaSave = true;
        int skill = 0;
        for(int x = 0; x<2; x++){
            if(x==0){
                skill = skill1;
            }else{
                skill = skill2;
            }
            switch(skill){
            case 1:
                this.arcana = true;
                break;
            case 2:
                this.deception = true;
                break;
            case 3:
                this.insight = true;
                break;
            case 4:
                this.intimidation = true;
                break;
            case 5:
                this.persuasion = true;
                break;
            case 6:
                this.religion = true;
                break;
            }
        }
        if(!this.equipment.equals("")){
            this.equipment += ", ";
        }
        this.equipment += item1 + item2 + item3 + ", two daggers";
        String originAppend = "";//choose an origin either wild magic or a
type of dragon
        switch(origin){
        case 0:
            originAppend ="Wild Magic, Wild Magic Surge, Tides of Chaos";
            break;
        case 1:
            originAppend ="Draconic Ancestry(Black), Wild Magic Surge,
Tides of Chaos";
            break;
        case 2:
            originAppend ="Draconic Ancestry(Blue), Wild Magic Surge,
Tides of Chaos";
            break;
        case 3:
            originAppend ="Draconic Ancestry(Brass), Wild Magic Surge,
Tides of Chaos";
            break;
        case 4:
            originAppend ="Draconic Ancestry(Bronze), Wild Magic Surge,
Tides of Chaos";
            break;
        }
    }
}

```

```

                break;
        case 5:
            originAppend = "Draconic Ancestry(Copper), Wild Magic Surge,
Tides of Chaos";
            break;
        case 6:
            originAppend = "Draconic Ancestry(Gold), Wild Magic Surge,
Tides of Chaos";
            break;
        case 7:
            originAppend = "Draconic Ancestry(Green), Wild Magic Surge,
Tides of Chaos";
            break;
        case 8:
            originAppend = "Draconic Ancestry(Red), Wild Magic Surge, Tides
of Chaos";
            break;
        case 9:
            originAppend = "Draconic Ancestry(Silver), Wild Magic Surge,
Tides of Chaos";
            break;
        case 10:
            originAppend = "Draconic Ancestry(White), Wild Magic Surge,
Tides of Chaos";
            break;

        }
        this.features += ", Spellcasting, Sorcerous origin: " +
originAppend;
    }
//for a completely commented character setter, see line 836
    public void warlock(int skill1, int skill2, String item1, String item2,
String item3, String item4, int patron){
    this.Class = "Warlock";
    this.hitDice = level +"D8";
    this.hitPoints = 8 + processing.calcModifier(Con);
    if(!this.otherProficiencies.equals("")){
        this.otherProficiencies += ", ";
    }
    this.otherProficiencies += "Light armour, simple weapons";
    this.wisSave = true;
    this.chaSave = true;
    int skill = 0;
    for(int x = 0; x<2; x++){
        if(x==0){
            skill = skill1;
        }else{
            skill = skill2;
        }
        switch(skill){
            case 1:
                this.arcana = true;
                break;
            case 2:
                this.deception = true;
                break;
            case 3:
                this.history = true;

```

```

        break;
    case 4:
        this.intimidation = true;
        break;
    case 5:
        this.investigation = true;
        break;
    case 6:
        this.nature = true;
        break;
    case 7:
        this.religion = true;
        break;
    }
}
if(!this.equipment.equals("")){
    this.equipment += ", ";
}
this.equipment += item1 + item2 + item3 + item4 +", leather armour,
two daggers";
this.armour = 2;
switch(patron){//choose what being the warlock gets their powers
from
    case 0:
        this.features += ", Otherworldly Patron (Archfey), Fey
presence";
        this.spells += "faerie fire, sleep";
        break;
    case 1:
        this.features += ", Otherworldly Patron (Fiend), Dark One's
Blessing";
        this.spells += "burning hands, command";
        break;
    case 2:
        this.features += ", Otherworldly Patron (Great Old One),
Awakening Mind";
        this.spells += "dissonant whispers, Tasha's hideous laughter";
        break;
    }
this.features += ", Pact magic";
this.spellAbil = "Charisma";
this.spellDC = 8 + this.profBonus +
processing.calcModifier(this.Cha);
this.spellMod = this.profBonus + processing.calcModifier(this.Cha);
}
//for a completely commented character setter, see line 836
public void wizard(int skill1, int skill2, String item1, String item2,
String item3){
    this.Class = "Wizard";
    this.hitDice = level +"D6";
    this.hitPoints = 6 + processing.calcModifier(Con);
    if(!this.otherProficiencies.equals("")){
        this.otherProficiencies += ", ";
    }
    this.otherProficiencies += "daggers, darts, slings, quaterstaffs,
light crossbows";
    this.intSave = true;
    this.wisSave = true;
}

```

```

int skill = 0;
for(int x = 0; x<2; x++){
    if(x==0){
        skill = skill1;
    }else{
        skill = skill2;
    }
    switch(skill){
        case 1:
            this.arcana = true;
            break;
        case 2:
            this.history = true;
            break;
        case 3:
            this.insight = true;
            break;
        case 4:
            this.investigation = true;
            break;
        case 5:
            this.medicine = true;
            break;
        case 6:
            this.religion = true;
            break;
    }
}
if(!this.equipment.equals("")){
    this.equipment += ", ";
}
this.equipment += item1 + item2 + item3 + ", spellbook";
this.features += ", Spellcasting, Arcane Recovery";
this.spellAbil = "Intelligence";
this.spellDC = 8 + this.profBonus +
processing.calcModifier(this.Int);
    this.spellMod = this.profBonus + processing.calcModifier(this.Int);
}
//end of classes

public void chooseProf(int prof) {//given an integer from one to 18 become
proficient in the relevant skill, a helper for small choices that give proficiency
switch (prof) {
    case 1:
        this.athletics = true;
        break;
    case 2:
        this.acrobatics = true;
        break;
    case 3:
        this.sleightOfHand = true;
        break;
    case 4:
        this.stealth = true;
        break;
    case 5:
        this.arcana = true;
        break;
}

```

```

        case 6:
            this.history = true;
            break;
        case 7:
            this.investigation = true;
            break;
        case 8:
            this.nature = true;
            break;
        case 9:
            this.religion = true;
            break;
        case 10:
            this.animalHandling = true;
            break;
        case 11:
            this.insight = true;
            break;
        case 12:
            this.medicine = true;
            break;
        case 13:
            this.perception = true;
            break;
        case 14:
            this.survival = true;
            break;
        case 15:
            this.deception = true;
            break;
        case 16:
            this.intimidation = true;
            break;
        case 17:
            this.performance = true;
            break;
        case 18:
            this.persuasion = true;
            break;
    }
}

//setter methods for ability scores

public void seta1(int strIn) {
    this.Str = this.Str + strIn;
}//set strength score

public void seta2(int dexIn) {
    this.Dex = this.Dex + dexIn;
}//set dexterity score

public void seta3(int conIn) {
    this.Con =this.Con + conIn;
}//set constitution score

public void seta4(int intIn) {
    this.Int = this.Int + intIn;
}

```

```

} //set intelligence score

public void seta5(int wisIn) {
    this.Wis = this.Wis + wisIn;
} //set wisdom score

public void seta6(int chaIn) {
    this.Cha = this.Cha + chaIn;
} //set charisma score
public void reset() { //reset all changes made by picking a race before a
race is selected incase the user went back. A method used in testing
    this.race = "";
    this.size = "medium";

    this.Str = 0;
    this.Dex = 0;
    this.Con = 0;
    this.Int = 0;
    this.Wis = 0;
    this.Cha = 0;

    //proficiencies
    this.athletics = false;
    this.acrobatics = false;
    this.sleightOfHand = false;
    this.stealth = false;
    this.arcana = false;
    this.history = false;
    this.investigation = false;
    this.nature = false;
    this.religion = false;
    this.animalHandling = false;
    this.insight = false;
    this.medicine = false;
    this.perception = false;
    this.survival = false;
    this.deception = false;
    this.intimidation = false;
    this.performance = false;
    this.persuasion = false;

    this.vision = "";
    this.languages = "";

    this.features = "";
    this.feats = "";
}
//after all choices are made this calculates all values that need
calculating
public void processingFinal(){
    //str
    this.StrMod = processing.calcModifier(this.Str); //calculates
modifier in the processing class
    this.strSaveV = this.StrMod; //sets the save to the relevant modifier
    if(this.strSave){
        this.strSaveV += this.profBonus; //add proficiency bonus if
proficient

```

```

        }
        this.athleticsV = this.StrMod;//athletics depends on the strength
modifier
        if(this.athletics){
            this.athleticsV += this.profBonus;//if proficient add two
            if(this.doubleAthletics){
                this.athleticsV += this.profBonus;//if doubled
add prof bonus again
            }
        }
//dex see line 1992 for in depth commenting
this.DexMod = processing.calcModifier(this.Dex);//calculate dex mod
and save
        this.dexSaveV = this.DexMod;
        if(this.dexSave){
            this.dexSaveV += this.profBonus;
        }
        this.acrobaticsV = this.DexMod;//calculate skills based on dexterity
        if(this.acrobatics){
            this.acrobaticsV += this.profBonus;
            if(this.doubleAcrobatics){
                this.acrobaticsV += this.profBonus;
            }
        }
        this.sleightOfHandV = this.DexMod;
        if(this.sleightOfHand){
            this.sleightOfHandV += this.profBonus;
            if(this.doubleSleight){
                this.sleightOfHandV += this.profBonus;
            }
        }
        this.stealthV = this.DexMod;
        if(this.stealth){
            this.stealthV += this.profBonus;
            if(this.doubleStealth){
                this.stealthV += this.profBonus;
            }
        }
//con see line 1992 for in depth commenting
this.ConMod = processing.calcModifier(this.Con);//no skills based on
constitution
        this.conSaveV = this.ConMod;
        if(this.conSave){
            this.conSaveV += this.profBonus;
        }
//int see line 1992 for in depth commenting
this.IntMod = processing.calcModifier(this.Int);//calculate skills
based on intelligence
        this.intSaveV = this.IntMod;
        if(this.intSave){
            this.intSaveV += this.profBonus;
        }
        this.arcanaV = this.IntMod;
        if(this.arcana){
            this.arcanaV += this.profBonus;
            if(this.doubleArcana){
                this.arcanaV += this.profBonus;
            }
        }

```

```

        }
        this.historyV = this.IntMod;
        if(this.history){
            this.historyV += this.profBonus;
            if(this.doubleHistory){
                this.historyV += this.profBonus;
            }
        }
        this.investigationV = this.IntMod;
        if(this.investigation){
            this.investigationV += this.profBonus;
            if(this.doubleInvestigation){
                this.investigationV += this.profBonus;
            }
        }
        this.natureV = this.IntMod;
        if(this.nature){
            this.natureV += this.profBonus;
            if(this.doubleNature){
                this.natureV += this.profBonus;
            }
        }
        this.religionV = this.IntMod;
        if(this.religion){
            this.religionV += this.profBonus;
            if(this.doubleReligion){
                this.religionV += this.profBonus;
            }
        }
    }
    //wis see line 1992 for in depth commenting
    this.WisMod = processing.calcModifier(this.Wis); //calculate wisdom
skills
    this.wisSaveV = this.WisMod;
    if(this.wisSave){
        this.wisSaveV += this.profBonus;
    }
    this.animalHandlingV = this.WisMod;
    if(this.animalHandling){
        this.animalHandlingV += this.profBonus;
        if(this.doubleAnimal){
            this.animalHandlingV += this.profBonus;
        }
    }
    this.insightV = this.WisMod;
    if(this.insight){
        this.insightV += this.profBonus;
        if(this.doubleInsight){
            this.insightV += this.profBonus;
        }
    }
    this.medicineV = this.WisMod;
    if(this.medicine){
        this.medicineV += this.profBonus;
        if(this.doubleMedicine){
            this.medicineV += this.profBonus;
        }
    }
    this.perceptionV = this.WisMod;

```

```

        if(this.perception){
            this.perceptionV += this.profBonus;
            if(this.doublePerception){
                this.perceptionV += this.profBonus;
            }
        }
        this.survivalV = this.WisMod;
        if(this.survival){
            this.survivalV += this.profBonus;
            if(this.doubleSurvival){
                this.survivalV += this.profBonus;
            }
        }
        //cha see line 1992 for in depth commenting
        this.ChaMod = processing.calcModifier(this.Cha); //calculate charisma
skills
        this.chaSaveV = this.ChaMod;
        if(this.chaSave){
            this.chaSaveV += this.profBonus;
        }
        this.deceptionV = this.ChaMod;
        if(this.deception){
            this.deceptionV += this.profBonus;
            if(this.doubleDeception){
                this.deceptionV += this.profBonus;
            }
        }
        this.intimidationV = this.ChaMod;
        if(this.intimidation){
            this.intimidationV += this.profBonus;
            if(this.doubleIntimidation){
                this.intimidationV += this.profBonus;
            }
        }
        this.performanceV = this.ChaMod;
        if(this.performance){
            this.performanceV += this.profBonus;
            if(this.doublePerformance){
                this.performanceV += this.profBonus;
            }
        }
        this.persuasionV = this.ChaMod;
        if(this.persuasion){
            this.persuasionV += this.profBonus;
            if(this.doublePersuasion){
                this.persuasionV += this.profBonus;
            }
        }
        this.initiative = this.DexMod; //initiative bonus equals dexterity
modifier
        this.passivePerception = 10 + this.perceptionV;
        switch(this.armour){ //takes reference set by armour int and gives
the appropriate armour class
            //no armour
            case 0:
                this.ac = 10 + this.DexMod;
                break;
            //light armour

```

```

case 1://padded
    this.ac = 11 + this.DexMod;
    this.stealthDis = true;
    break;
case 2://leather
    this.ac = 11 + this.DexMod;
    break;
case 3://studded
    this.ac = 12 + this.DexMod;
    break;
//medium
case 11://hide
    if(this.DexMod >= 2){//add dex modifier to a maximum of 2
        this.ac = 12 + 2;
    }else{
        this.ac = 12 + this.DexMod;
    }
    break;
case 12://hide
    if(this.DexMod >= 2){
        this.ac = 13 + 2;
    }else{
        this.ac = 13 + this.DexMod;
    }
    break;
case 13://scale
    if(this.DexMod >= 2){
        this.ac = 14 + 2;
    }else{
        this.ac = 14 + this.DexMod;
    }
    this.stealthDis = true;//gives disadvantage to stealth rolls
    break;
case 14://breastplate
    if(this.DexMod >= 2){
        this.ac = 14 + 2;
    }else{
        this.ac = 14 + this.DexMod;
    }
    break;
case 15://half plate
    if(this.DexMod >= 2){
        this.ac = 15 + 2;
    }else{
        this.ac = 15 + this.DexMod;
    }
    this.stealthDis = true;
    break;
//heavy
case 21://ringmail
    this.ac = 14;
    this.stealthDis = true;
    break;
case 22://chainmail
    this.ac = 16;
    this.stealthDis = true;
    break;
case 23://splint

```

```

        this.ac = 17;
        this.stealthDis = true;
        break;
    case 24://ringmail
        this.ac = 18;
        this.stealthDis = true;
        break;
    case 31:
        this.ac = 10 + this.DexMod + this.ConMod;//barabarian
unarmoured defence
        break;
    case 32:
        this.ac = 10 + this.DexMod + this.WisMod;//monk unarmoured
defence
        break;
    }
    if(this.shield){
        this.ac += 2;//add +2 ac for shield
    }
    if(this.defence){
        this.ac++; //add one ac for feat
    }
}
public static void main(String args[]) {
    //main method
}
}

```

## Processing.java

```
package cc;

import java.util.Random;
//all useful methods for processing needed values, such as generating random
numbers.

public class processing {
    public static long calcModifier(long scoreIn){
        long mod = (scoreIn - 10)/2;//-10 and divide by 2
        //needs to always round down, the following if's round down negative
values
        if(scoreIn == 9){
            mod = -1;
        }
        if(scoreIn == 7){
            mod = -2;
        }
        if(scoreIn == 5){
            mod = -3;
        }
        if(scoreIn == 3){
            mod = -4;
        }//Ability scores cannot get lower than 3
        return mod;
    }//takes an ability score and turns it into a modifier

    public static int rand4D6() {// 4d6
        Random rand = new Random();
        int FourD6;
        int D61 = rand.nextInt((6 - 1) + 1); //generates 4 random numbers
between 1 and 6
        int D62 = rand.nextInt((6 - 1) + 1) + 1;
        int D63 = rand.nextInt((6 - 1) + 1) + 1;
        int D64 = rand.nextInt((6 - 1) + 1) + 1;
        int y = D61;//sets y to equal the first d6 value
        if (y > D62) {
            y = D62;
        }//if the second value is lower than the first, y equals the lower
value
        if (y > D63) {
            y = D63;
        }//sets y equal to the lower value
        if (y > D64) {
            y = D64;
        }//finally y is equal to the lowest value
        FourD6 = D61 + D62 + D63 + D64 - y;//adds all the values and
subtracts the lowest number, ignoring it
        return FourD6;//returns 4d6 minus the lowest result
    }
    public static int oneD4() {
        Random rand = new Random();
        int randomNum = rand.nextInt((4 - 1) + 1) + 1;
        return randomNum;
    }//returns a roll of one 4 sided dice
    public static int twoD4() {
        Random rand = new Random();
        int randomNum = rand.nextInt((4 - 2) + 1) + 2;
        return randomNum;
    }
}
```

```
//returns a roll of two four sided dice
public static int oneD6() {
    Random rand = new Random();
    int randomNum = rand.nextInt((6 - 1) + 1) + 1;
    return randomNum;
}//returns a roll of one six sided dice
public static int twoD6() {
    Random rand = new Random();
    int randomNum = rand.nextInt((6 - 2) + 1) + 2;
    return randomNum;
}//returns a roll of two six sided dice
public static int twoD8() {
    Random rand = new Random();
    int randomNum = rand.nextInt((8 - 2) + 1) + 2;
    return randomNum;
}//returns a roll of two eight sided dice
public static int twoD10() {
    Random rand = new Random();
    int randomNum = rand.nextInt((10 - 2) + 1) + 2;
    return randomNum;
}//returns a roll of two ten sided dice
}
```

## Helper package

### Login.java

```
package helper;

import com.google.appengine.api.datastore.DatastoreService;
import com.google.appengine.api.datastore.DatastoreServiceFactory;
import com.google.appengine.api.datastore.Entity;
import com.google.appengine.api.datastore.EntityNotFoundException;
import com.google.appengine.api.datastore.FetchOptions;
import com.google.appengine.api.datastore.Key;
import com.google.appengine.api.datastore.KeyFactory;
import com.google.appengine.api.datastore.PreparedQuery;
import com.google.appengine.api.datastore.Query;
import com.google.appengine.api.datastore.Query.Filter;
import com.google.appengine.api.datastore.Query.FilterOperator;
import com.google.appengine.api.datastore.Query.FilterPredicate;
import com.google.appengine.api.datastore.Query.CompositeFilter;
import com.google.appengine.api.datastore.Query.CompositeFilterOperator;
//import app engine tools
import java.io.IOException;
import java.util.Date;
import java.util.List;

import cc.coursework;//imports the main character variable holder class called
coursework, allows a coursework object to be used
//holds all the methods for appending and altering the datastore
public class login {//class method

    public login(){
        //constructor
    }
    public boolean checkLogin(String usernameIn, String passwordIn){//checks if
the users username and password are correct
        DatastoreService ds =
DatastoreServiceFactory.getDatastoreService(); //allows the cloud database to be
used
        Filter sieve = new FilterPredicate("username",
FilterOperator.EQUAL, usernameIn); //creates a filter that finds the given
//username in the username property
        Query checkUser = new Query("login").setFilter(sieve); //implements
the filter on the login kind
        Entity user = ds.prepare(checkUser).asSingleEntity(); //creates a
user holding the received entity from the query
        if(user == null){ //if the username is not found by the jquery
            System.out.println("username not found");
            return false; //return the username doesn't exist
        }
        if(passwordIn.equals(user.getProperty("password"))){ //checks if the
password provided is identical to the password of that username
            return true; //return username and password correct
        }
        return false; //return false, password wrong
    }
}
```

```

public boolean nameFree(String usernameIn){//check if username is free for
account creation
    DatastoreService ds =
DatastoreServiceFactory.getDatastoreService(); // use cloud database
    Filter sieve = new FilterPredicate("username",
FilterOperator.EQUAL, usernameIn); //filter to get that username
    Query checkUser = new Query("login").setFilter(sieve); //implements
the filter on the login kind
    Entity user = ds.prepare(checkUser).asSingleEntity(); //create a user
holding that username
    if(user == null){//username doesnt exist
        System.out.println("username not found"); //username can be
used
        return true;
    }
    return false; //username taken and cant be used
}
public coursework getCharacter(String ownerIn, int saveIn){ // get an object
of the coursework class holding all saved choices
    coursework pc = new coursework(); //create an instance of the
coursework class
    DatastoreService ds =
DatastoreServiceFactory.getDatastoreService(); //datastore
    Filter sieve = new FilterPredicate("save slot",
FilterOperator.EQUAL, ownerIn + saveIn); //filter to get the saveslot
    //save slot is a unique key combining the username and save slot
    Query checkUser = new Query("character").setFilter(sieve); //search
the character kind for save slots
    Entity player = ds.prepare(checkUser).asSingleEntity(); //temporary
entity
    if(player == null){ //if save slot doesnt exist
        System.out.println("no character's found"); //alert
        return null;
    }else{ //set all the needed values from the datastore to the object
to be returned
        try{ //try catch blocks for error validation
            pc.pName = (String) player.getProperty("characterName"); //sets
the objects name to equal the value in the store
            pc.Class = (String) player.getProperty("class");
            pc.background = (String) player.getProperty("background");
            pc.name = (String) player.getProperty("realName");
            pc.race = (String) player.getProperty("race");
            pc.alignment = (String) player.getProperty("alignment");
            pc.xp = (long) player.getProperty("xp");
        }catch(Exception e){
            System.out.println(e + "1"); //if thrown exception
output the error and the number of the block
        }try{
            pc.Str = (long) player.getProperty("strengthScore"); //datastore
saves as longs so all values needed to be converted to long
            pc.Dex = (long) player.getProperty("dexterityScore");
            pc.Con = (long) player.getProperty("constitutionScore");
            pc.Int = (long) player.getProperty("intelligenceScore");
            pc.Wis = (long) player.getProperty("wisdomScore");
            pc.Cha = (long) player.getProperty("charismaScore");
        }catch(Exception e){
            System.out.println(e + "2"); //section 2 error
        }try{

```

```

pc.StrMod = (long)player.getProperty("strengthMod");
pc.DexMod = (long)player.getProperty("dexterityMod");
pc.ConMod = (long)player.getProperty("constitutionMod");
pc.IntMod = (long)player.getProperty("intelligenceMod");
pc.WisMod = (long)player.getProperty("wisdomMod");
pc.ChaMod = (long)player.getProperty("charismaMod");
}catch(Exception e){
    System.out.println(e + "3");//section 3 error
}try{
pc.strSaveV = (long)player.getProperty("strengthSave");
pc.dexSaveV = (long)player.getProperty("dexteritySave");
pc.conSaveV = (long)player.getProperty("constitutionSave");
pc.intSaveV = (long)player.getProperty("intelligenceSave");
pc.wisSaveV = (long)player.getProperty("wisdomSave");
pc.chaSaveV = (long)player.getProperty("charismaSave");

pc.strSave = (boolean)player.getProperty("strengthSaveB");
pc.dexSave = (boolean)player.getProperty("dexteritySaveB");
pc.conSave = (boolean)player.getProperty("constitutionSaveB");
pc.intSave = (boolean)player.getProperty("intelligenceSaveB");
pc.wisSave = (boolean)player.getProperty("wisdomSaveB");
pc.chaSave = (boolean)player.getProperty("charismaSaveB");
}catch(Exception e){
    System.out.println(e + "4");//section 4 error
}try{
pc.ac = (long)player.getProperty("AC");
pc.initiative = (long)player.getProperty("initiative");
pc.speed = (long)player.getProperty("speed");
pc.hitPoints = (long)player.getProperty("HP");
pc.hitDice = (String)player.getProperty("hit dice");
}catch(Exception e){
    System.out.println(e + "5");//section 5 error
}try{
    pc.spellAbil = (String)player.getProperty("spellAbil");
    pc.spellDC = (Long)player.getProperty("spellDC");//if
not a psllcasting should be 111
    pc.spellMod = (Long)player.getProperty("spellMod");
}catch(Exception e){
    System.out.println(e + "6");//section 6 error, not all
classes have spells
}try{
pc.personality =
(String)player.getProperty("personalityTraits");
pc.ideal = (String)player.getProperty("ideal");
pc.bond = (String)player.getProperty("bond");
pc.flaw = (String)player.getProperty("flaw");
}catch(Exception e){
    System.out.println(e + "7");//section 7 eroor
}try{
pc.acrobaticsV = (long)player.getProperty("acrobatics");
pc.animalHandlingV =
(long)player.getProperty("animalHandling");
pc.arcanaV = (long)player.getProperty("arcana");
pc.athleticsV = (long)player.getProperty("athletics");
pc.deceptionV = (long)player.getProperty("deception");
pc.historyV = (long)player.getProperty("history");
pc.insightV = (long)player.getProperty("insight");
pc.intimidationV = (long)player.getProperty("intimidation");

```

```

pc.investigationV = (long)player.getProperty("investigation");
pc.medicineV = (long)player.getProperty("medicine");
pc.natureV = (long)player.getProperty("nature");
pc.perceptionV = (long)player.getProperty("perception");
pc.performanceV = (long)player.getProperty("performance");
pc.persuasionV = (long)player.getProperty("persuasion");
pc.religionV = (long)player.getProperty("religion");
pc.sleightOfHandV = (long)player.getProperty("sleightOfHand");
pc.stealthV = (long)player.getProperty("stealth");
pc.survivalV = (long)player.getProperty("survival");
pc.passivePerception =
(long)player.getProperty("passivePerception");

pc.acrobatics = (boolean)player.getProperty("acrobaticsb");
pc.animalHandling =
(boolean)player.getProperty("animalHandlingb");
pc.arcana = (boolean)player.getProperty("arcanab");
pc athletics = (boolean)player.getProperty("athleticsb");
pc.deception = (boolean)player.getProperty("deceptionb");
pc.history = (boolean)player.getProperty("historyb");
pc.insight = (boolean)player.getProperty("insightb");
pc.intimidation =
(boolean)player.getProperty("intimidationb");
pc.investigation =
(boolean)player.getProperty("investigationb");
pc.medicine = (boolean)player.getProperty("medicineb");
pc.nature = (boolean)player.getProperty("natureb");
pc.perception = (boolean)player.getProperty("perceptionb");
pc.performance = (boolean)player.getProperty("performanceb");
pc.persuasion = (boolean)player.getProperty("persuasionb");
pc.religion = (boolean)player.getProperty("religionb");
pc.sleightOfHand =
(boolean)player.getProperty("sleightOfHandb");
pc.stealth = (boolean)player.getProperty("stealthb");
pc.survival = (boolean)player.getProperty("survivalb");
}catch(Exception e){
    System.out.println(e + "8");//section 8 error
}try{
    pc.languages = (String)player.getProperty("languages");
    pc.otherProficiencies =
(String)player.getProperty("otherProficiencies");
    pc.money = (long)player.getProperty("gold");
    pc.equipment = (String)player.getProperty("equipment");
    pc.features = (String)player.getProperty("features");
}catch(Exception e){
    System.out.println(e + "9");//section 9 error
}try{
    pc.name = (String)player.getProperty("playerName");
    pc.age = (long)player.getProperty("age");
    pc.realH = (String)player.getProperty("height");
    pc.realW = (String)player.getProperty("weight");
    pc.eyes = (String)player.getProperty("eyes");
    pc.skin = (String)player.getProperty("skin");
    pc.hair = (String)player.getProperty("hair");
    pc.backstory = (String)player.getProperty("backstory");
    pc.vision = (String)player.getProperty("vision");
}catch(Exception e){
    System.out.println(e + "10");//section 10 error
}

```

```

        }
    }
    return pc;//return the instanced class with all its values
}

//when character creation is finished, all the coursework classes
attributes are saved to the datastore
public void createCharacter(String usernameIn, coursework p, Integer
slot){//accepts the created character, username and save slot
    DatastoreService ds =
DatastoreServiceFactory.getDatastoreService();//uses datastore
    Filter sieve = new FilterPredicate("username",
FilterOperator.EQUAL, usernameIn);//finds usernames of that username
    Query checkUser = new Query("login").setFilter(sieve);//applies the
filter to get the entity
    Entity user = ds.prepare(checkUser).asSingleEntity();//set the
entity to the one found by the query
    if(user == null){//username doesnt exist, shouldn't occur
        System.out.println("username not found on creation");//alert
    }else{//if user does exist
        System.out.println("char created");//alert
        Filter charactersieve = new FilterPredicate("save slot",
FilterOperator.EQUAL, usernameIn + slot);//search saveslots for the chosen save
slot
        //save slot is a concatenation of the username and save
number, which creates a unique key
        Query checkcharacter = new
Query("character").setFilter(charactersieve);
        Entity character =
ds.prepare(checkcharacter).asSingleEntity();
        if(character == null){//if slot isn't used
            character = new Entity("character");//create a new
character
        }
        character.setProperty("save slot", usernameIn + slot);//append
save slot if it didn't already exist
        character.setProperty("characterName", p.pName);//set
properties for each of the objects attributes
        character.setProperty("class", p.Class);
        character.setProperty("background", p.background);
        character.setProperty("realName", p.name);
        character.setProperty("race", p.race);
        character.setProperty("alignment", p.alignment);
        character.setProperty("xp", p.xp);

        character.setProperty("strengthScore", p.Str);
        character.setProperty("dexterityScore", p.Dex);
        character.setProperty("constitutionScore", p.Con);
        character.setProperty("intelligenceScore", p.Int);
        character.setProperty("wisdomScore", p.Wis);
        character.setProperty("charismaScore", p.Cha);

        character.setProperty("strengthMod", p.StrMod);
        character.setProperty("dexterityMod", p.DexMod);
        character.setProperty("constitutionMod", p.ConMod);
        character.setProperty("intelligenceMod", p.IntMod);
        character.setProperty("wisdomMod", p.WisMod);
        character.setProperty("charismaMod", p.ChaMod);
}

```

```

character.setProperty("strengthSave", p.strSaveV);
character.setProperty("dexteritySave", p.dexSaveV);
character.setProperty("constitutionSave", p.conSaveV);
character.setProperty("intelligenceSave", p.intSaveV);
character.setProperty("wisdomSave", p.wisSaveV);
character.setProperty("charismaSave", p.chaSaveV);

character.setProperty("strengthSaveB", p.strSave);
character.setProperty("dexteritySaveB", p.dexSave);
character.setProperty("constitutionSaveB", p.conSave);
character.setProperty("intelligenceSaveB", p.intSave);
character.setProperty("wisdomSaveB", p.wisSave);
character.setProperty("charismaSaveB", p.chaSave);

character.setProperty("AC", p.ac);
character.setProperty("initiative", p.initiative);
character.setProperty("speed", p.speed);
character.setProperty("HP", p.hitPoints);
character.setProperty("hit dice", p.hitDice);
character.setProperty("xp", p.xp);

character.setProperty("spellAbil", p.spellAbil);
character.setProperty("spellDC", p.spellDC);
character.setProperty("spellMod", p.spellMod);

character.setProperty("personalityTraits", p.personality);
character.setProperty("ideal", p.ideal);
character.setProperty("bond", p.bond);
character.setProperty("flaw", p.flaw);

character.setProperty("acrobatics", p.acrobaticsV);
character.setProperty("animalHandling", p.animalHandlingV);
character.setProperty("arcana", p.arcanaV);
character.setProperty("athletics", p.athleticsV);
character.setProperty("deception", p.deceptionV);
character.setProperty("history", p.historyV);
character.setProperty("insight", p.insightV);
character.setProperty("intimidation", p.intimidationV);
character.setProperty("investigation", p.investigationV);
character.setProperty("medicine", p.medicineV);
character.setProperty("nature", p.natureV);
character.setProperty("perception", p.perceptionV);
character.setProperty("performance", p.performanceV);
character.setProperty("persuasion", p.persuasionV);
character.setProperty("religion", p.religionV);
character.setProperty("sleightOfHand", p.sleightOfHandV);
character.setProperty("stealth", p.stealthV);
character.setProperty("survival", p.survivalV);
character.setProperty("passivePerception",
p.passivePerception);

character.setProperty("acrobaticsb", p.acrobatics);
character.setProperty("animalHandlingb", p.animalHandling);
character.setProperty("arcanab", p.arcana);
character.setProperty("athleticsb", p.athletics);
character.setProperty("deceptionb", p.deception);

```

```

        character.setProperty("historyb", p.history);
        character.setProperty("insightb", p.insight);
        character.setProperty("intimidationb", p.intimidation);
        character.setProperty("investigationb", p.investigation);
        character.setProperty("medicineb", p.medicine);
        character.setProperty("natureb", p.nature);
        character.setProperty("perceptionb", p.perception);
        character.setProperty("performanceb", p.performance);
        character.setProperty("persuasionb", p.persuasion);
        character.setProperty("religionb", p.religion);
        character.setProperty("sleightOfHandb", p.sleightOfHand);
        character.setProperty("stealthb", p.stealth);
        character.setProperty("survivalb", p.survival);

        character.setProperty("languages", p.languages);
        character.setProperty("otherProficiencies",
p.otherProficiencies);
        character.setProperty("gold", p.money);
        character.setProperty("equipment", p.equipment);
        character.setProperty("features", p.features);

        character.setProperty("playerName", p.name);
        character.setProperty("age", p.age);
        character.setProperty("height", p.realH);
        character.setProperty("weight", p.realW);
        character.setProperty("eyes", p.eyes);
        character.setProperty("skin", p.skin);
        character.setProperty("hair", p.hair);
        character.setProperty("backstory", p.backstory);
        character.setProperty("vision", p.vision);
        ds.put(character); //put in the datastore
    }
}
}

```

## login

### Index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

<jsp:useBean id="helper" class="helper.Login" scope="page" /><!-- allows the
helper datastore class to be used -->
<!-the login page for users to access their accounts -->
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login</title><!-- sets title -->
</head>
<body onLoad="document.login.username.focus();"><!-- focuses on the username field
-->

<%
    String u = "";//username variable
    String p = "";//password variable

    try {//attempts to get the username and password and check if the
account exists
        if(request.getParameter("username") != null &&
request.getParameter("password") != null
            //check the username and password has been submitted
            && request.getParameter("username") != "" &&
request.getParameter("password") != ""){
            //check the username and password isnt blank
            u = request.getParameter("username");//set temporary username
and password variables
            p = request.getParameter("password");
            if (helper.checkLogin(u, p)) {//call helper function to check
if that username and password exist in the datastore
                session.setAttribute("username", u);//set the username
as an attribute so the rest of the program knows the user
                response.sendRedirect("race.jsp");//as the username and
password are correct send the user to the first page of character creation
            } else {//need username/password
                out.println("your username or password is
incorrect");//incorrect username or password message
            }
        }
    }

} catch (Exception e) {//catch error
    out.println("problem in getting u and p"+e);//print error to
the system
}

%>

<form name="Login"><!-- form for jsp processing -->
<div id = "align"><!-- centered signup details -->
    <h2>Signup Details</h2> <!-- header -->
        <br /><div> Username:<input type="text" name="username"
class="input"></div> <!-- input text fields -->
```

```
        <br /><div style = "position: relative; left: 1.5px">
Password:<input type="password" name="password" class="input"></div> <!-- input
password field -->
        <br /> <input type="submit" value="Submit" class="input"><!--
submit button -->
        <a class = "align" href="/account.jsp">Create account</a><!-- link
to create account page -->
    </div>
</form>

</body>
</html>
```

```

account.jsp
<%@ page import="com.google.appengine.api.datastore.DatastoreServiceFactory"%>
<%@ page import="com.google.appengine.api.datastore.DatastoreService"%>
<%@ page import="com.google.appengine.api.datastore.Query"%>
<%@ page import="com.google.appengine.api.datastore.Entity"%>
<%@ page import="com.google.appengine.api.datastore.FetchOptions"%>
<%@ page import="com.google.appengine.api.datastore.Key"%>
<%@ page import="com.google.appengine.api.datastore.KeyFactory"%>
<%@ page import="com.google.appengine.api.datastore.Query.Filter"%>
<%@ page import="com.google.appengine.api.datastore.Query.FilterOperator"%>
<%@ page import="com.google.appengine.api.datastore.Query.FilterPredicate"%>
<%@ page import="com.google.appengine.api.datastore.Query.CompositeFilter"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%><!-- import all appspot -->
<jsp:useBean id="helper" class="helper.Login" scope="page" /><!-- create a Bean
for the helper class for datastore usage -->
<!--localhost:8888/_ah/admin/datastore url for local datastore when testing-->
<!--the web page for adding wanted accounts to the Datastore to then log in to. -->
<!DOCTYPE html>
<html>
<script type="text/javascript"></script>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css"><!-- load css page
-->
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login</title>
</head>
<body onLoad="document.login.username.focus();"><!-- focus selection on username
panel -->

<%
    String u = "";//initialise all variables for holding user username
and password choices
    String p1 = "";
    String p2 = "";

    try {//catch errors
        if(request.getParameter("username") != null &&
request.getParameter("password1")
            != null && request.getParameter("password2") !=
null){
            if(request.getParameter("username") != "" &&
request.getParameter("password1")!= ""
                && request.getParameter("password2") !=
""){//check that values have been entered
                u = request.getParameter("username");//set initialised
vairables to values put in the fields
                p1 = request.getParameter("password1");
                p2 = request.getParameter("password2");
                if(helper.nameFree(u)){//use the helper bean to check if the
name is already on the datastore
                    if (p1.equals(p2)) {//check if the two entered passwords are
identical
                        DatastoreService datastore =
DatastoreServiceFactory.getDatastoreService();//standard datastore functions
                        Key loginKey = KeyFactory.createKey("login", "id");
                        Entity user = new Entity("login");//create a new entity
for saving username and password

```

```

        Entity character = new Entity("character");//create a
new entity for saving characters to this account
        user.setProperty("username", u);//add the username
entered to the entity under the username column
        user.setProperty("password", p1);//add the password to
the entity under the password column
        datastore.put(user);//add the entity to the datastore
        character.setProperty("owner", u);//add the username to
the owner column of the character entity
        datastore.put(character);//add the entity to the
datastore
        response.sendRedirect("index.jsp");//redirect the user
to the login page
    } else {//need username/password
        out.println("your passwords are not identical");
    }
} else{//password is taken
    out.println("The username: " + u + " is taken");
}
}
}
}
}
} catch (Exception e) {
    System.out.println("error " + e);//output unexpected error to
console
}
%>

```

```

<h2 style =text-align:center>create account</h2>
<form method="post" name="Login">
<div id = "align">
<br /><div style = "position: relative; right:
1.5px">Username:<input type="text" name="username"></div>
<br /><div>Password:<input type="password"
name="password1"></div><!-- style attributes adjust the location of the textboxes
aligning them -->
<br /><div style = "position: relative; right: 33px">Re-enter
Password:<input type="password" name="password2" ></div>
<br /><input type="submit" value="Submit">
</div>
</form>
<!-- standard form for entering a username and two entries of a
password -->
</body>
</html>

```

## Race selection

### race.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--page for choosing the race of the coursework object and using the proper setter
method. --&gt;
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;script type="text/javascript"&gt;&lt;/script&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;script src="radio.js"&gt;&lt;/script&gt;
<!-- import the dynamic radio buttons and choice information --&gt;
&lt;script src="valid.js"&gt;&lt;/script&gt;

&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;%
    coursework player = new coursework(); //instances the coursework
class so that an object for the character can be made to set the values of
//initialised and declared variables for holding choices
    int selectedTool = 0;
    int selectedAncestry = 0;
    int selectedLanguage = 0;
    int selectedprof = 0;
    int abil1 = 0;
    int abil2 = 0;
    int prof1 = 0;
    int prof2 = 0;
    String tool = null;
    String lang = null;
    String ancestor = null;
    int prof = 1;
        int selectedItem = 0;
        if (request.getParameter("race") != null) {//on form
            submission get the value of the race, so the correct setter method can be run
                selectedItem =
Integer.parseInt(request.getParameter("race")); //convert the string value to a
useable integer
        }

        switch(selectedItem){//switch the selected race integer
            case 0://if 0 the race is a high elf
                if (request.getParameter("language") != null)
//if a high elf then language radio buttons have been created
                    selectedLanguage =
Integer.parseInt(request.getParameter("language")); //get chosen language from new
radio buttons
                        switch (selectedLanguage) { //switch chosen
language integer
                            case 0://the 0 option was Dwarvish
                                so the parameter for the High elf setter method is set to dwarf
                                    lang ="Dwarvish";
                                break;
</pre>
```

```

        case 1://etc
            lang = "Giant";
            break;
        case 2:
            lang = "Gnomish";
            break;
        case 3:
            lang = "Goblin";
            break;
        case 4:
            lang = "Halfling";
            break;
        case 5:
            lang = "Orc";
            break;
        case 6:
            lang = "Abyssal";
            break;
        case 7:
            lang = "Celestial";
            break;
        case 8:
            lang = "Draconic";
            break;
        case 9:
            lang = "Deep Speech";
            break;
        case 10:
            lang = "Infernal";
            break;
        case 11:
            lang = "Primordial";
            break;
        case 12:
            lang = "Sylvan";
            break;
        case 13:
            lang = "Undercommon";
            break;

    }
    player.reset(); //reset character choices
incase the user returns to this page and rechooses race
    player.HighElf(lang); //setter method for
high elf race with language parameters
    session.setAttribute("player1",
player); //set the object of user choices as a session variable for use on later
pages

    response.sendRedirect("nameAppearance.jsp"); //redirect to the next page for
further choices
}

break;
case 1://wood elf, no parameters
    player.reset(); //reset choices
    player.WoodElf(); //wood elf setter method

```

```

        session.setAttribute("player1", player); //set object as
session variable
        response.sendRedirect("nameAppearance.jsp"); //redirect
break;
    case 2://dark elf, no parameters
        player.reset(); //standard setter, session and redirect
        player.DarkElf();
        session.setAttribute("player1", player);
        response.sendRedirect("nameAppearance.jsp");
        break;
    case 3://hill dwarf
        if (request.getParameter("tool") != null) {
            selectedTool =
Integer.parseInt(request.getParameter("tool")); //get integer from created tool
radio buttons
            switch (selectedTool) { //switch on choice
                case 0:
                    tool = "smith's tools"; //set
parameters depending on choice
                    break;
                case 1:
                    tool = "brewer's supplies";
                    break;
                case 2:
                    tool = "mason's tools";
                    break;
            }
            player.reset(); //reset
            player.HillDwarf(tool); //setter method
with chosen tool as parameter
            session.setAttribute("player1",
player); //set object as session variable
            response.sendRedirect("nameAppearance.jsp"); //redirect to next page
        }
        break;
    case 4://mountain dwarf, same as hill dwarf with a different
setter
        if (request.getParameter("tool") != null) {
            selectedTool =
Integer.parseInt(request.getParameter("tool"));
            switch (selectedTool) {
                case 0:
                    tool = "smith's tools";
                    break;
                case 1:
                    tool = "brewer's supplies";
                    break;
                case 2:
                    tool = "mason's tools";
                    break;
            }
            player.reset();
            player.MountainDwarf(tool); //mountain
dwarf setter
            session.setAttribute("player1", player);

```

```

        response.sendRedirect("nameAppearance.jsp");
    }
    break;
case 5://lightfoot halfling, no parameters
    player.reset();
    player.LightfootHalfling();
    session.setAttribute("player1", player);
    response.sendRedirect("nameAppearance.jsp");
    break;
case 6://Stout halfling, no parameters
    player.reset();
    player.StoutHalfling();
    session.setAttribute("player1", player);
    response.sendRedirect("nameAppearance.jsp");
    break;
case 7: //human
    if (request.getParameter("langH") != null) {
        selectedLanguage =
Integer.parseInt(request.getParameter("langH")); //get chosen language from
generated radio buttons
        switch (selectedLanguage) {
            case 0://depending on chosen radio button
set set the language parameter
                lang = "Dwarvish";
                break;
            case 1:
                lang = "Elven";
                break;
            case 2:
                lang = "Giant";
                break;
            case 3:
                lang = "Gnomish";
                break;
            case 4:
                lang = "Goblin";
                break;
            case 5:
                lang = "Halfling";
                break;
            case 6:
                lang = "Orc";
                break;
            case 7:
                lang = "Abyssal";
                break;
            case 8:
                lang = "Celestial";
                break;
            case 9:
                lang = "Draconic";
                break;
            case 10:
                lang = "Deep Speech";
                break;
            case 11:
                lang = "Infernal";

```

```

        break;
    case 12:
        lang = "Primordial";
        break;
    case 13:
        lang = "Sylvan";
        break;
    case 14:
        lang = "Undercommon";
        break;
    }
player.reset();
player.Human(lang); //set with language parameter
session.setAttribute("player1", player);
response.sendRedirect("nameAppearance.jsp");
}
break;
case 8://dragonborn
if (request.getParameter("dragon") != null) {
selectedAncestry =
Integer.parseInt(request.getParameter("dragon")); //get selected radio button
choice for switch
set the parameter to black
switch (selectedAncestry) {
    case 0://if the black button is selected
        ancestor ="black";
        break;
    case 1://etc
        ancestor = "blue";
        break;
    case 2:
        ancestor = "brass";
        break;
    case 3:
        ancestor = "bronze";
        break;
    case 4:
        ancestor = "copper";
        break;
    case 5:
        ancestor = "gold";
        break;
    case 6:
        ancestor = "green";
        break;
    case 7:
        ancestor = "red";
        break;
    case 8:
        ancestor = "silver";
        break;
    case 9:
        ancestor = "white";
        break;
}
player.reset();
player.Dragonborn(ancestor); //set to dragonborn
with parameter

```

```

        session.setAttribute("player1", player);
        response.sendRedirect("nameAppearance.jsp");
    }
    break;
case 9://forest gnome,no parameters
    player.reset();
    player.ForestGnome();
    session.setAttribute("player1", player);
    response.sendRedirect("nameAppearance.jsp");
    break;
case 10://rock gnome, no parameters
    player.reset();
    player.RockGnome();
    session.setAttribute("player1", player);
    response.sendRedirect("nameAppearance.jsp");
    break;
case 11://halfelf, many parameters
if (request.getParameter("language") != null) {
    selectedLanguage =
Integer.parseInt(request.getParameter("language")); //get chosen language
    switch (selectedLanguage) {
case 0:
    lang = "Dwarvish"; //set language parameter
    break;
case 1:
    lang = "Giant";
    break;
case 2:
    lang = "Gnomish";
    break;
case 3:
    lang = "Goblin";
    break;
case 4:
    lang = "Halfling";
    break;
case 5:
    lang = "Orc";
    break;
case 6:
    lang = "Abyssal";
    break;
case 7:
    lang = "Celestial";
    break;
case 8:
    lang = "Draconic";
    break;
case 9:
    lang = "Deep Speech";
    break;
case 10:
    lang = "Infernal";
    break;
case 11:
    lang = "Primordial";
    break;
case 12:

```

```

        lang = "Sylvan";
        break;
    case 13:
        lang = "Undercommon";
        break;
    }
try{
    if (request.getParameter("abils") != null) {
        String[] a = request.getParameterValues("abils");//get
all choices from ability score checkboxes
        abil1 = Integer.parseInt(a[0]);//set first ability
score number as first value in array
        abil2 = Integer.parseInt(a[1]);//set second ability
score as second value in array
        if (request.getParameter("profs") != null) {//get all
proficiency choices
            String[] b = request.getParameterValues("profs");
            prof1 = Integer.parseInt(b[0]);//set profession 1
to first selected checkbox
            prof2 = Integer.parseInt(b[1]);//set profession 2
to second checkbox
            player.reset();
            player.HalfElf(abil1, abil2, prof1, prof2,
lang);//set to half elf with all parameters
            session.setAttribute("player1", player);
            response.sendRedirect("nameAppearance.jsp");
        }
    }
}catch(Exception e){
}
}
break;

case 12://half orc, no parameters
player.reset();
player.HalfOrc();
session.setAttribute("player1", player);
response.sendRedirect("nameAppearance.jsp");
break;
case 13://tiefling, no parameters
player.reset();
player.Tiefling();
session.setAttribute("player1", player);
response.sendRedirect("nameAppearance.jsp");
break;
case 14://human variant
if (request.getParameter("langH") != null) {
    selectedLanguage =
Integer.parseInt(request.getParameter("langH"));//get chosen language
    switch (selectedLanguage) {
        case 0://set parameter to chosen language
            lang ="Dwarvish";
            break;
        case 1:
            lang = "Elven";
            break;
        case 2:
            lang = "Giant";

```

```

        break;
    case 3:
        lang = "Gnomish";
        break;
    case 4:
        lang = "Goblin";
        break;
    case 5:
        lang = "Halfling";
        break;
    case 6:
        lang = "Orc";
        break;
    case 7:
        lang = "Abyssal";
        break;
    case 8:
        lang = "Celestial";
        break;
    case 9:
        lang = "Draconic";
        break;
    case 10:
        lang = "Deep Speech";
        break;
    case 11:
        lang = "Infernal";
        break;
    case 12:
        lang = "Primordial";
        break;
    case 13:
        lang = "Sylvan";
        break;
    case 14:
        lang = "Undercommon";
        break;
    }
    try{
        if (request.getParameter("abil") != null) {
            String[] a = request.getParameterValues("abil");//get chosen
ability scores
            abil1 = Integer.parseInt(a[0]);//set abil 1 to first checkbox
            abil2 = Integer.parseInt(a[1]);//set abil 2 to second checkbox
            if (request.getParameter("prof") != null) {
                selectedprof =
                    Integer.parseInt(request.getParameter("prof"));//get proficiency int, which goes
                    to a switch in the setter
                player.reset();
                player.HumanVar(lang, abil1, abil2, selectedprof,
lang);//human var setter with parameters
                session.setAttribute("player1", player);
                response.sendRedirect("nameAppearance.jsp");
            }
        }
    }catch(Exception e){
    }
}

```

```

        break;
    }
%>
<div id="navcontainer">
    <!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <!-- contains links to the main pages -->
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>

<h1>Bram's character creator</h1>
<h2>Race</h2>
<form>
    <div align="center">
        <!-- align centrally -->
        <input type="radio" name="race" id="hElf" value="0"
onclick="highElf();">High elf<!-- radio buttons for the choice of each race -->
        <input type="radio" name="race" id="wElf" value="1"
onclick="woodElf();">Wood elf
        <!-- the have integer values for the switch case block -->
        <input type="radio" name="race" id="dElf" value="2"
onclick="darkElf();">Dark elf <br>
        <!-- runs functions from the radio js page -->
        <input type="radio" name="race" id="hDwarf" value="3"
onclick="dwarfTool();hill();">Hill dwarf<!-- each function populates lower divs
with information and more choices -->
        <input type="radio" name="race" id="mDwarf" value="4"
onclick="dwarfTool();mount();">Mountain dwarf <br>
        <input type="radio" name="race" id="lHalf" value="5"
onclick="lightfoot();">Lightfoot halfling
        <input type="radio" name="race" id="sHalf" value="6"
onclick="stout();">Stout halfling <br>
        <input type="radio" name="race" id="human" value="7"
onclick="hum();">Human
        <input type="radio" name="race" id="humanVar" value="14"
onclick="humVar();checkboxlimit(abil,2);">Human (variant)<br><!-- calls
checkboxlimit from the valid js page, this limits the number of checkboxes that
can be checked -->
        <input type="radio" name="race" id="drag" value="8"
onclick="ancestry();">Dragonborn<br>
        <input type="radio" name="race" id="fGnome" value="9"
onclick="fGno();">Forest gnome
        <input type="radio" name="race" id="rGnome" value="10"
onclick="rGno();">Rock gnome <br>
        <input type="radio" name="race" id="halfE" value="11"
onclick="halfelf();checkboxlimit(abil,2);checkboxlimit(profs,2);">Half-elf<br>
        <input type="radio" name="race" id="halfO" value="12"
onclick="orc();">Half-orc<br>
        <input type="radio" name="race" id="tief" value="13"
onclick="tiefling();">Tiefling<br>
    </div>
    <h3 id="head" align="center"></h3>
    <div id="radioDiv" align="center"></div>
    <!-- div for holding radio buttons for further choices -->
    <h3 id="head2" align="center"></h3>

```

```
<div id="checkDiv" align="center"></div>
<!-- div for checkboxes -->
<h3 id="head3" align="center"></h3>
<div id="profDiv" align="center"></div>
<!-- div for holding information on what proficiencies that race
gets -->
<h3>information:</h3>
<div id="infoBox" align="center">
<!-- div for generic race information -->
</div>
<div align="center">
    <input type="submit">
</div>
</form>
</body>
</html>
```

## radio.js

```
//all the functions for the race.jsp page allowing the dynamic creation of radio buttons. For all see the first hElf function for comments

function hElf(){//high elf info function
    document.getElementById("radioDiv").innerHTML = "";//reset div's that may already hold information or buttons
    document.getElementById("checkDiv").innerHTML = "";
    document.getElementById("profDiv").innerHTML = "";
    document.getElementById("head").innerHTML = "";
    document.getElementById("head").innerHTML = "choose a language";//inform of choices
    document.getElementById("head2").innerHTML = "";
    document.getElementById("head3").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "High elves are the most magically talented of the elves <br>" +
        " they learn an additional language and begin with a magical cantrip. They have improved dexterity and intelligence.";
    //set info
    var objDiv = document.getElementById("radioDiv");//create a useable variable equal to the radioDiv div.

    var radioItem1 = document.createElement("input");//create radio items that are radio buttons with the name language and values
    radioItem1.type = "radio";
    radioItem1.name = "language";
    radioItem1.value = "0";//value for switch case block

    var radioItem2 = document.createElement("input");//each radio item is created seperately in this way
    radioItem2.type = "radio";
    radioItem2.name = "language";
    radioItem2.value = "1";

    var radioItem3 = document.createElement("input");
    radioItem3.type = "radio";
    radioItem3.name = "language";
    radioItem3.value = "2";

    var radioItem4 = document.createElement("input");
    radioItem4.type = "radio";
    radioItem4.name = "language";
    radioItem4.value = "3";

    var radioItem5 = document.createElement("input");
    radioItem5.type = "radio";
    radioItem5.name = "language";
    radioItem5.value = "4";

    var radioItem6 = document.createElement("input");
    radioItem6.type = "radio";
    radioItem6.name = "language";
    radioItem6.value = "5";

    var radioItem7 = document.createElement("input");
    radioItem7.type = "radio";
    radioItem7.name = "language";
```

```

radioItem7.value = "6";

var radioItem8 = document.createElement("input");
radioItem8.type = "radio";
radioItem8.name = "language";
radioItem8.value = "7";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "language";
radioItem9.value = "8";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "language";
radioItem10.value = "9";

var radioItem11 = document.createElement("input");
radioItem11.type = "radio";
radioItem11.name = "language";
radioItem11.value = "10";

var radioItem12 = document.createElement("input");
radioItem12.type = "radio";
radioItem12.name = "language";
radioItem12.value = "11";

var radioItem13 = document.createElement("input");
radioItem13.type = "radio";
radioItem13.name = "language";
radioItem13.value = "12";

var radioItem14 = document.createElement("input");
radioItem14.type = "radio";
radioItem14.name = "language";
radioItem14.value = "13";

var objTextNode1 = document.createTextNode("Dwarvish");//create a text
node to say what language the radio button is to the user
var objTextNode2 = document.createTextNode("Giant");
var objTextNode3 = document.createTextNode("Gnomish");
var objTextNode4 = document.createTextNode("Goblin");
var objTextNode5 = document.createTextNode("Halfling");
var objTextNode6 = document.createTextNode("Orc");
var objTextNode7 = document.createTextNode("Abyssal");
var objTextNode8 = document.createTextNode("Celestial");
var objTextNode9 = document.createTextNode("Draconic");
var objTextNode10 = document.createTextNode("Deep Speech");
var objTextNode11 = document.createTextNode("Infernal");
var objTextNode12 = document.createTextNode("Primordial");
var objTextNode13 = document.createTextNode("Sylvan");
var objTextNode14 = document.createTextNode("Undercommon");

var objLabel = document.createElement("label");//create labels
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);//append the radio button to the label
objLabel.appendChild(objTextNode1);//append the text to the radio button

```

```

var objLabel2 = document.createElement("label");//append all others in the
same way
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;

```

```

objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

objDiv.appendChild(objLabel); //append labels to the original div
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);

}

function wElf(){//wood elf function
document.getElementById("radioDiv").innerHTML = "";//clear and set info
document.getElementById("checkDiv").innerHTML = "";
document.getElementById("profDiv").innerHTML = "";
document.getElementById("head").innerHTML = "";
document.getElementById("head2").innerHTML = "";
document.getElementById("head3").innerHTML = "";
document.getElementById("infoBox").innerHTML = "";
document.getElementById("infoBox").innerHTML = "an elf that lives in a
forest <br> they have improved speed, dexterity and wisdom, they are proficient
in perception";
}
function dElf(){//dark elf
document.getElementById("radioDiv").innerHTML = "";//clear and set info
document.getElementById("checkDiv").innerHTML = "";
document.getElementById("profDiv").innerHTML = "";
document.getElementById("head").innerHTML = "";
document.getElementById("head2").innerHTML = "";
document.getElementById("head3").innerHTML = "";
document.getElementById("infoBox").innerHTML = "";
document.getElementById("infoBox").innerHTML = "An elf that dwells
underground <br> they have improved dexterity, charisma and night vision, as well
as proficiency with many exotic weapons";
}
function dwarfTool(){//clear and set base dwarf info
document.getElementById("radioDiv").innerHTML = "";
document.getElementById("checkDiv").innerHTML = "";
document.getElementById("profDiv").innerHTML = "";
document.getElementById("head").innerHTML = "";
document.getElementById("head").innerHTML = "choose a tool proficiency";
document.getElementById("head2").innerHTML = "";
}

```

```

        document.getElementById("head3").innerHTML = "";
        document.getElementById("infoBox").innerHTML = "";
        var objDiv = document.getElementById("radioDiv");//standard radio button
creation with 3 options for tools
        var radioItem1 = document.createElement("input");
radioItem1.type = "radio";
radioItem1.name = "tool";
radioItem1.value = "0";

        var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "tool";
radioItem2.value = "1";

        var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "tool";
radioItem3.value = "2";

        var objTextNode1 = document.createTextNode("smith's tools");
var objTextNode2 = document.createTextNode("brewer's supplies");
var objTextNode3 = document.createTextNode("mason's tools");

        var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

        var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

        var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

        objDiv.appendChild(objLabel);
        objDiv.appendChild(objLabel2);
        objDiv.appendChild(objLabel3);

    }

function hill(){//hill dwarf info
    document.getElementById("infoBox").innerHTML = "a dwarf from the hills <br>
they have improved constitution and wisdom and gain proficiency with a toolset of
your choice";
}
function mount(){//mountain dwarf info
    document.getElementById("infoBox").innerHTML = "a dwarf from the
mountains<br> they have improved constitution and strength and gain proficiency
with a toolset of your choice";
}

function lightfoot(){//lightfoot halfling
    document.getElementById("radioDiv").innerHTML = "";//clear and set info
    document.getElementById("checkDiv").innerHTML = "";
    document.getElementById("profDiv").innerHTML = "";
}

```

```

document.getElementById("head").innerHTML = "";
document.getElementById("head2").innerHTML = "";
document.getElementById("head3").innerHTML = "";
document.getElementById("infoBox").innerHTML = "";
document.getElementById("infoBox").innerHTML = "a quick hobbit <br> with
improved dexterity, charisma and an ability to hide";
}
function stout(){//stout halfling
document.getElementById("radioDiv").innerHTML = "";//clear and set info
document.getElementById("checkDiv").innerHTML = "";
document.getElementById("profDiv").innerHTML = "";
document.getElementById("head").innerHTML = "";
document.getElementById("head2").innerHTML = "";
document.getElementById("head3").innerHTML = "";
document.getElementById("infoBox").innerHTML = "";
document.getElementById("infoBox").innerHTML = "a slow hobbit <br> with
improved dexterity, constitution and resistance to poison and disease";
}

function hum(){//human
document.getElementById("radioDiv").innerHTML = "";//clear and set info
document.getElementById("checkDiv").innerHTML = "";
document.getElementById("profDiv").innerHTML = "";
document.getElementById("head").innerHTML = "";
document.getElementById("head").innerHTML = "choose a language";
document.getElementById("head2").innerHTML = "";
document.getElementById("head3").innerHTML = "";
document.getElementById("infoBox").innerHTML = "";
document.getElementById("infoBox").innerHTML = "the standard human with all
stats improved and a choice of language";
var objDiv = document.getElementById("radioDiv");//create language radio
buttons

var radioItem1 = document.createElement("input");
radioItem1.type = "radio";
radioItem1.name = "langH";
radioItem1.value = "0";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "langH";
radioItem2.value = "1";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "langH";
radioItem3.value = "2";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "langH";
radioItem4.value = "3";

var radioItem5 = document.createElement("input");
radioItem5.type = "radio";
radioItem5.name = "langH";
radioItem5.value = "4";

```

```

var radioItem6 = document.createElement("input");
radioItem6.type = "radio";
radioItem6.name = "langH";
radioItem6.value = "5";

var radioItem7 = document.createElement("input");
radioItem7.type = "radio";
radioItem7.name = "langH";
radioItem7.value = "6";

var radioItem8 = document.createElement("input");
radioItem8.type = "radio";
radioItem8.name = "langH";
radioItem8.value = "7";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "langH";
radioItem9.value = "8";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "langH";
radioItem10.value = "9";

var radioItem11 = document.createElement("input");
radioItem11.type = "radio";
radioItem11.name = "langH";
radioItem11.value = "10";

var radioItem12 = document.createElement("input");
radioItem12.type = "radio";
radioItem12.name = "langH";
radioItem12.value = "11";

var radioItem13 = document.createElement("input");
radioItem13.type = "radio";
radioItem13.name = "langH";
radioItem13.value = "12";

var radioItem14 = document.createElement("input");
radioItem14.type = "radio";
radioItem14.name = "langH";
radioItem14.value = "13";

var radioItem15 = document.createElement("input");
radioItem15.type = "radio";
radioItem15.name = "langH";
radioItem15.value = "14";

var objTextNode1 = document.createTextNode("Dwarvish");
var objTextNode2 = document.createTextNode("Elvish");
var objTextNode3 = document.createTextNode("Giant");
var objTextNode4 = document.createTextNode("Gnomish");

```

```

var objTextNode5 = document.createTextNode("Goblin");
var objTextNode6 = document.createTextNode("Halfling");
var objTextNode7 = document.createTextNode("Orc");
var objTextNode8 = document.createTextNode("Abyssal");
var objTextNode9 = document.createTextNode("Celestial");
var objTextNode10 = document.createTextNode("Draconic");
var objTextNode11 = document.createTextNode("Deep Speech");
var objTextNode12 = document.createTextNode("Infernal");
var objTextNode13 = document.createTextNode("Primordial");
var objTextNode14 = document.createTextNode("Sylvan");
var objTextNode15 = document.createTextNode("Undercommon");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);

```

```

objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

objDiv.appendChild(objLabel);
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
}

function humVar(){//human variant
    hum(); //run human function to clear and set all info
    document.getElementById("infoBox").innerHTML = "a variant wher the user
gets to choose" +
        " 2 impprooved ability scores, a proficiency, a language and a trait"; //add
all variant human info
    document.getElementById("head2").innerHTML = "choose your 2 improved
ability scores";
}

```

```

var objDiv = document.getElementById("checkDiv");//create inputs in the
checkbox div
var radioItem1 = document.createElement("input");
radioItem1.type = "checkbox";//checkbox
radioItem1.name = "abil";//ability scores
radioItem1.value = "1";

var radioItem2 = document.createElement("input");
radioItem2.type = "checkbox";
radioItem2.name = "abil";
radioItem2.value = "2";

var radioItem3 = document.createElement("input");
radioItem3.type = "checkbox";
radioItem3.name = "abil";
radioItem3.value = "3";

var radioItem4 = document.createElement("input");
radioItem4.type = "checkbox";
radioItem4.name = "abil";
radioItem4.value = "4";

var radioItem5 = document.createElement("input");
radioItem5.type = "checkbox";
radioItem5.name = "abil";
radioItem5.value = "5";

var radioItem6 = document.createElement("input");
radioItem6.type = "checkbox";
radioItem6.name = "abil";
radioItem6.value = "6";

var objTextNode1 = document.createTextNode("Strength");//set text to the types
of ability scores
var objTextNode2 = document.createTextNode("Dexterity");
var objTextNode3 = document.createTextNode("Constitution");
var objTextNode4 = document.createTextNode("Intelligence");
var objTextNode5 = document.createTextNode("Wisdom");
var objTextNode6 = document.createTextNode("Charisma");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);

```

```

objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);

var objDiv = document.getElementById("profDiv"); //use the proficiency div
document.getElementById("head3").innerHTML = "Choose a proficiency:";

var radioItem1 = document.createElement("input"); //create radio buttons for
proficiencies in the prof div
radioItem1.type = "radio";
radioItem1.name = "prof";
radioItem1.value = "1";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "prof";
radioItem2.value = "2";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "prof";
radioItem3.value = "3";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "prof";
radioItem4.value = "4";

var radioItem5 = document.createElement("input");
radioItem5.type = "radio";
radioItem5.name = "prof";
radioItem5.value = "5";

var radioItem6 = document.createElement("input");
radioItem6.type = "radio";
radioItem6.name = "prof";
radioItem6.value = "6";

var radioItem7 = document.createElement("input");
radioItem7.type = "radio";
radioItem7.name = "prof";
radioItem7.value = "7";

```

```

var radioItem8 = document.createElement("input");
radioItem8.type = "radio";
radioItem8.name = "prof";
radioItem8.value = "8";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "prof";
radioItem9.value = "9";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "prof";
radioItem10.value = "10";

var radioItem11 = document.createElement("input");
radioItem11.type = "radio";
radioItem11.name = "prof";
radioItem11.value = "11";

var radioItem12 = document.createElement("input");
radioItem12.type = "radio";
radioItem12.name = "prof";
radioItem12.value = "12";

var radioItem13 = document.createElement("input");
radioItem13.type = "radio";
radioItem13.name = "prof";
radioItem13.value = "13";

var radioItem14 = document.createElement("input");
radioItem14.type = "radio";
radioItem14.name = "prof";
radioItem14.value = "14";

var radioItem15 = document.createElement("input");
radioItem15.type = "radio";
radioItem15.name = "prof";
radioItem15.value = "15";

var radioItem16 = document.createElement("input");
radioItem16.type = "radio";
radioItem16.name = "prof";
radioItem16.value = "16";

var radioItem17 = document.createElement("input");
radioItem17.type = "radio";
radioItem17.name = "prof";
radioItem17.value = "17";

var radioItem18 = document.createElement("input");
radioItem18.type = "radio";
radioItem18.name = "prof";
radioItem18.value = "18";

var objTextNode1 = document.createTextNode("athletics");//text for each of the
proficiency options

```

```

var objTextNode2 = document.createTextNode("acrobatics");
var objTextNode3 = document.createTextNode("sleight of hand");
var objTextNode4 = document.createTextNode("stealth");
var objTextNode5 = document.createTextNode("arcana");
var objTextNode6 = document.createTextNode("history");
var objTextNode7 = document.createTextNode("investigation");
var objTextNode8 = document.createTextNode("nature");
var objTextNode9 = document.createTextNode("religion");
var objTextNode10 = document.createTextNode("animal handling");
var objTextNode11 = document.createTextNode("insight");
var objTextNode12 = document.createTextNode("medicine");
var objTextNode13 = document.createTextNode("perception");
var objTextNode14 = document.createTextNode("survival");
var objTextNode15 = document.createTextNode("deception");
var objTextNode16 = document.createTextNode("intimidation");
var objTextNode17 = document.createTextNode("performance");
var objTextNode18 = document.createTextNode("persuasion");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

```

```
var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem3.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

var objLabel16 = document.createElement("label");
objLabel16.htmlFor = radioItem16.id;
objLabel16.appendChild(radioItem16);
objLabel16.appendChild(objTextNode16);

var objLabel17 = document.createElement("label");
objLabel17.htmlFor = radioItem17.id;
objLabel17.appendChild(radioItem17);
objLabel17.appendChild(objTextNode17);

var objLabel18 = document.createElement("label");
objLabel18.htmlFor = radioItem18.id;
objLabel18.appendChild(radioItem18);
objLabel18.appendChild(objTextNode18);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
```

```

objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
objDiv.appendChild(objLabel16);
objDiv.appendChild(objLabel17);
objDiv.appendChild(objLabel18);
}

function ancestry(){//dragonborn
    document.getElementById("radioDiv").innerHTML = "";//clear and set info
    document.getElementById("checkDiv").innerHTML = "";
    document.getElementById("profDiv").innerHTML = "";
    document.getElementById("head").innerHTML = "";
    document.getElementById("head").innerHTML = "choose your type of dragon";
    document.getElementById("head2").innerHTML = "";
    document.getElementById("head3").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "A humanoid dragon with
improved strength and charisma. <br> they also get to choose a type of dragon they
are related to, this changes what breath attack they have and what types of damage
they are immune to";
    var objDiv = document.getElementById("radioDiv");//add radio buttons to
radioDiv
    var radioItem1 = document.createElement("input");//radio buttons for
draconic ancestor colour
    radioItem1.type = "radio";
    radioItem1.name = "dragon";
    radioItem1.value = "0";

    var radioItem2 = document.createElement("input");
    radioItem2.type = "radio";
    radioItem2.name = "dragon";
    radioItem2.value = "1";

    var radioItem3 = document.createElement("input");
    radioItem3.type = "radio";
    radioItem3.name = "dragon";
    radioItem3.value = "2";

    var radioItem4 = document.createElement("input");
    radioItem4.type = "radio";
    radioItem4.name = "dragon";
    radioItem4.value = "3";

    var radioItem5 = document.createElement("input");
    radioItem5.type = "radio";
    radioItem5.name = "dragon";
    radioItem5.value = "4";

    var radioItem6 = document.createElement("input");
    radioItem6.type = "radio";
    radioItem6.name = "dragon";
    radioItem6.value = "5";
}

```

```

var radioItem7 = document.createElement("input");
radioItem7.type = "radio";
radioItem7.name = "dragon";
radioItem7.value = "6";

var radioItem8 = document.createElement("input");
radioItem8.type = "radio";
radioItem8.name = "dragon";
radioItem8.value = "7";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "dragon";
radioItem9.value = "8";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "dragon";
radioItem10.value = "9";

var objTextNode1 = document.createTextNode("Black");//coulour choice text
var objTextNode2 = document.createTextNode("Blue");
var objTextNode3 = document.createTextNode("Brass");
var objTextNode4 = document.createTextNode("Bronze");
var objTextNode5 = document.createTextNode("Copper");
var objTextNode6 = document.createTextNode("Gold");
var objTextNode7 = document.createTextNode("Green");
var objTextNode8 = document.createTextNode("Red");
var objTextNode9 = document.createTextNode("Silver");
var objTextNode10 = document.createTextNode("White");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

```

```

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
objDiv.appendChild(objLabel16);
objDiv.appendChild(objLabel17);
objDiv.appendChild(objLabel18);
objDiv.appendChild(objLabel19);
objDiv.appendChild(objLabel10);
}

function fGno(){//forest gnome
    document.getElementById("radioDiv").innerHTML = "";//clear and set info
    document.getElementById("checkDiv").innerHTML = "";
    document.getElementById("profDiv").innerHTML = "";
    document.getElementById("head").innerHTML = "";
    document.getElementById("head2").innerHTML = "";
    document.getElementById("head3").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "a woodland gnome <br>they have improved intelligence and dexterity. They are also able to conjure small illusions";
}
function rGno(){//rock gnome
    document.getElementById("radioDiv").innerHTML = "";//clear and set info
    document.getElementById("checkDiv").innerHTML = "";
    document.getElementById("profDiv").innerHTML = "";
    document.getElementById("head").innerHTML = "";
    document.getElementById("head2").innerHTML = "";
    document.getElementById("head3").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "";
}

```

```

        document.getElementById("infoBox").innerHTML = "a mountaing gnome <br> they
have improved intelligence and constitution and are able to tinker and create
small clockwork devices";
}
function halfelf(){//half elf
    document.getElementById("radioDiv").innerHTML = "";//clear and set info
    document.getElementById("checkDiv").innerHTML = "";
    document.getElementById("profDiv").innerHTML = "";
    document.getElementById("head").innerHTML = "";
    document.getElementById("head2").innerHTML = "";
    document.getElementById("head3").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "a half man half elf
<br>they gain two improved skills and two proficiencies as well as a language";

    document.getElementById("head").innerHTML = "choose a language";

    var objDiv = document.getElementById("radioDiv");//add language radio
buttons to radioDiv

    var radioItem1 = document.createElement("input");
    radioItem1.type = "radio";
    radioItem1.name = "language";
    radioItem1.value = "0";

    var radioItem2 = document.createElement("input");
    radioItem2.type = "radio";
    radioItem2.name = "language";
    radioItem2.value = "1";

    var radioItem3 = document.createElement("input");
    radioItem3.type = "radio";
    radioItem3.name = "language";
    radioItem3.value = "2";

    var radioItem4 = document.createElement("input");
    radioItem4.type = "radio";
    radioItem4.name = "language";
    radioItem4.value = "3";

    var radioItem5 = document.createElement("input");
    radioItem5.type = "radio";
    radioItem5.name = "language";
    radioItem5.value = "4";

    var radioItem6 = document.createElement("input");
    radioItem6.type = "radio";
    radioItem6.name = "language";
    radioItem6.value = "5";

    var radioItem7 = document.createElement("input");
    radioItem7.type = "radio";
    radioItem7.name = "language";
    radioItem7.value = "6";

    var radioItem8 = document.createElement("input");
    radioItem8.type = "radio";
    radioItem8.name = "language";

```

```

radioItem8.value = "7";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "language";
radioItem9.value = "8";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "language";
radioItem10.value = "9";

var radioItem11 = document.createElement("input");
radioItem11.type = "radio";
radioItem11.name = "language";
radioItem11.value = "10";

var radioItem12 = document.createElement("input");
radioItem12.type = "radio";
radioItem12.name = "language";
radioItem12.value = "11";

var radioItem13 = document.createElement("input");
radioItem13.type = "radio";
radioItem13.name = "language";
radioItem13.value = "12";

var radioItem14 = document.createElement("input");
radioItem14.type = "radio";
radioItem14.name = "language";
radioItem14.value = "13";

var objTextNode1 = document.createTextNode("Dwarvish");//text
var objTextNode2 = document.createTextNode("Giant");
var objTextNode3 = document.createTextNode("Gnomish");
var objTextNode4 = document.createTextNode("Goblin");
var objTextNode5 = document.createTextNode("Halfling");
var objTextNode6 = document.createTextNode("Orc");
var objTextNode7 = document.createTextNode("Abyssal");
var objTextNode8 = document.createTextNode("Celestial");
var objTextNode9 = document.createTextNode("Draconic");
var objTextNode10 = document.createTextNode("Deep Speech");
var objTextNode11 = document.createTextNode("Infernal");
var objTextNode12 = document.createTextNode("Primordial");
var objTextNode13 = document.createTextNode("Sylvan");
var objTextNode14 = document.createTextNode("Undercommon");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");

```

```
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);
```

```

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);

document.getElementById("head2").innerHTML = "choose your 2 improved
ability scores";
var objDiv = document.getElementById("checkDiv");//checkDiv
var radioItem1 = document.createElement("input");//create checkboxes in the
checkDiv
radioItem1.type = "checkbox";
radioItem1.name = "abilis";
radioItem1.value = "1";

var radioItem2 = document.createElement("input");
radioItem2.type = "checkbox";
radioItem2.name = "abilis";
radioItem2.value = "2";

var radioItem3 = document.createElement("input");
radioItem3.type = "checkbox";
radioItem3.name = "abilis";
radioItem3.value = "3";

var radioItem4 = document.createElement("input");
radioItem4.type = "checkbox";
radioItem4.name = "abilis";
radioItem4.value = "4";

var radioItem5 = document.createElement("input");
radioItem5.type = "checkbox";
radioItem5.name = "abilis";
radioItem5.value = "5";

var objTextNode1 = document.createTextNode("Strength");//text
var objTextNode2 = document.createTextNode("Dexterity");
var objTextNode3 = document.createTextNode("Constitution");
var objTextNode4 = document.createTextNode("Intelligence");
var objTextNode5 = document.createTextNode("Wisdom");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");

```

```

objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);

var objDiv = document.getElementById("profDiv"); //profDiv
document.getElementById("head3").innerHTML = "Choose 2 proficiencies:";

var radioItem1 = document.createElement("input"); //create proficiency radio
buttons
radioItem1.type = "checkbox";
radioItem1.name = "profs";
radioItem1.value = "1";

var radioItem2 = document.createElement("input");
radioItem2.type = "checkbox";
radioItem2.name = "profs";
radioItem2.value = "2";

var radioItem3 = document.createElement("input");
radioItem3.type = "checkbox";
radioItem3.name = "profs";
radioItem3.value = "3";

var radioItem4 = document.createElement("input");
radioItem4.type = "checkbox";
radioItem4.name = "profs";
radioItem4.value = "4";

var radioItem5 = document.createElement("input");
radioItem5.type = "checkbox";
radioItem5.name = "profs";
radioItem5.value = "5";

var radioItem6 = document.createElement("input");
radioItem6.type = "checkbox";
radioItem6.name = "profs";
radioItem6.value = "6";

```

```

var radioItem7 = document.createElement("input");
radioItem7.type = "checkbox";
radioItem7.name = "profs";
radioItem7.value = "7";

var radioItem8 = document.createElement("input");
radioItem8.type = "checkbox";
radioItem8.name = "profs";
radioItem8.value = "8";

var radioItem9 = document.createElement("input");
radioItem9.type = "checkbox";
radioItem9.name = "profs";
radioItem9.value = "9";

var radioItem10 = document.createElement("input");
radioItem10.type = "checkbox";
radioItem10.name = "profs";
radioItem10.value = "10";

var radioItem11 = document.createElement("input");
radioItem11.type = "checkbox";
radioItem11.name = "profs";
radioItem11.value = "11";

var radioItem12 = document.createElement("input");
radioItem12.type = "checkbox";
radioItem12.name = "profs";
radioItem12.value = "12";

var radioItem13 = document.createElement("input");
radioItem13.type = "checkbox";
radioItem13.name = "profs";
radioItem13.value = "13";

var radioItem14 = document.createElement("input");
radioItem14.type = "checkbox";
radioItem14.name = "profs";
radioItem14.value = "14";

var radioItem15 = document.createElement("input");
radioItem15.type = "checkbox";
radioItem15.name = "profs";
radioItem15.value = "15";

var radioItem16 = document.createElement("input");
radioItem16.type = "checkbox";
radioItem16.name = "profs";
radioItem16.value = "16";

var radioItem17 = document.createElement("input");
radioItem17.type = "checkbox";
radioItem17.name = "profs";
radioItem17.value = "17";

var radioItem18 = document.createElement("input");
radioItem18.type = "checkbox";

```

```

radioItem18.name = "profs";
radioItem18.value = "18";

var objTextNode1 = document.createTextNode("athletics");//text
var objTextNode2 = document.createTextNode("acrobatics");
var objTextNode3 = document.createTextNode("sleight of hand");
var objTextNode4 = document.createTextNode("stealth");
var objTextNode5 = document.createTextNode("arcana");
var objTextNode6 = document.createTextNode("history");
var objTextNode7 = document.createTextNode("investigation");
var objTextNode8 = document.createTextNode("nature");
var objTextNode9 = document.createTextNode("religion");
var objTextNode10 = document.createTextNode("animal handling");
var objTextNode11 = document.createTextNode("insight");
var objTextNode12 = document.createTextNode("medicine");
var objTextNode13 = document.createTextNode("perception");
var objTextNode14 = document.createTextNode("survival");
var objTextNode15 = document.createTextNode("deception");
var objTextNode16 = document.createTextNode("intimidation");
var objTextNode17 = document.createTextNode("performance");
var objTextNode18 = document.createTextNode("persuasion");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

```

```

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem3.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

var objLabel16 = document.createElement("label");
objLabel16.htmlFor = radioItem16.id;
objLabel16.appendChild(radioItem16);
objLabel16.appendChild(objTextNode16);

var objLabel17 = document.createElement("label");
objLabel17.htmlFor = radioItem17.id;
objLabel17.appendChild(radioItem17);
objLabel17.appendChild(objTextNode17);

var objLabel18 = document.createElement("label");
objLabel18.htmlFor = radioItem18.id;
objLabel18.appendChild(radioItem18);
objLabel18.appendChild(objTextNode18);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);

```

```

        objDiv.appendChild(objLabel14);
        objDiv.appendChild(objLabel15);
        objDiv.appendChild(objLabel16);
        objDiv.appendChild(objLabel17);
        objDiv.appendChild(objLabel18);
        objDiv.appendChild(objLabel19);
        objDiv.appendChild(objLabel10);
        objDiv.appendChild(objLabel11);
        objDiv.appendChild(objLabel12);
        objDiv.appendChild(objLabel13);
        objDiv.appendChild(objLabel14);
        objDiv.appendChild(objLabel15);
        objDiv.appendChild(objLabel16);
        objDiv.appendChild(objLabel17);
        objDiv.appendChild(objLabel18);
    }

function orc(){//half orc
    document.getElementById("radioDiv").innerHTML = ""; //clear and set info
    document.getElementById("checkDiv").innerHTML = "";
    document.getElementById("profDiv").innerHTML = "";
    document.getElementById("head").innerHTML = "";
    document.getElementById("head2").innerHTML = "";
    document.getElementById("head3").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "a half man half orc <br>
they get improved strength, constitution and intimidation";
}
function tiefling(){//tiefling
    document.getElementById("radioDiv").innerHTML = "";//clear and set info
    document.getElementById("checkDiv").innerHTML = "";
    document.getElementById("profDiv").innerHTML = "";
    document.getElementById("head").innerHTML = "";
    document.getElementById("head2").innerHTML = "";
    document.getElementById("head3").innerHTML = "";
    document.getElementById("infoBox").innerHTML = "a half man half demon <br>
they get improved charisma, intelligence and resistance to fire";
}

```

### valid.js

```
//function that limits the number of checkboxes that can be checked.

function checkboxlimit(checkgroup, limit){//set limit of checkbox checks
    var checkgroup=checkgroup//make the checkgroup a useable variable
    var limit=limit//make the limit of the max number a useable variable
    for (var i=0; i<checkgroup.length; i++){//for each checkbox in the group
        checkgroup[i].onclick=function(){//run function onclick
            var checkedcount=0//variable, number of boxes that are checked
            for (var i=0; i<checkgroup.length; i++)//for each checkbox in the
group
                checkedcount+=(checkgroup[i].checked)? 1 : 0//if that checkbox
is checked increase the checkedcount by 1
                if (checkedcount>limit){//if checkcount is greater than the given
limit
                    this.checked=false//uncheck the box
                }
            }
        }
    }
}
```

## Background selection

```
nameAppearance.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--lets the player choose their background options for the background setter
method acting on their coursework object -->
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<style>
#center{
    margin-top: 12%;
    width: 4%;
    float: left;
}
</style>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<%
session.getAttribute("player1");//get coursework class holding choices
coursework player = (coursework)session.getAttribute("player1");//set an object to
equal the session variable
//variables initialised and declared to hold choices from the form
int age = 0;
String height = "";
int feet = 0;
int inches = 0;
int weight = 0;
boolean randHW = false;//whether or not the user wants a random weight and height
String rName = "";//real name
String pName = "";//player character name
String eyes = "";
String skin = "";
String hair = "";
boolean[] flags = new boolean[6];//array of flags for confirming if all choices
are made
boolean flag3 = false;
boolean flag4 = false;
boolean flag5 = false;
boolean bigFlag = false;
if (request.getParameter("age") != null) {//sets age
    if (request.getParameter("age") != "") {//checks an age was entered
        try{
            age = Integer.parseInt(request.getParameter("age"));//converts
input age to an integer
            flags[0] = true;//sets first flag to true
        }catch(Exception e){
            //catches if non integers are entered
        }
    }
}
if (request.getParameter("rName") != null) {//sets real name
    if (request.getParameter("rName") != ""){

```

```

        rName = request.getParameter("rName");//sets the name to the
entered name
            flags[1] = true;//sets he second flag to true
        }
    }
if (request.getParameter("cName") != null) {//sets player character name
    if (request.getParameter("cName") != ""){
        pName = request.getParameter("cName");
        flags[2] = true;
    }
}
if (request.getParameter("eyes") != null) {//set eye information
    if (request.getParameter("eyes") != ""){
        eyes = request.getParameter("eyes");
        flags[3] = true;
    }
}
if (request.getParameter("skin") != null) {//sets skin information
    if (request.getParameter("skin") != ""){
        skin = request.getParameter("skin");
        flags[4] = true;
    }
}
if (request.getParameter("hair") != null) {//sets hair information
    if (request.getParameter("hair") != ""){
        hair = request.getParameter("hair");
        flags[5] = true;
    }
}
if (request.getParameter("rand") != null) {//sets random height and weight flag
    int rand = Integer.parseInt(request.getParameter("rand"));
    if(rand == 1){//if the box was checked
        randHW = true;//set random height/weight to true
    }
}

if (request.getParameter("feet") != null) {//set the number of feet in the palyers
height
    if (request.getParameter("feet") != ""){
        try{
            feet = Integer.parseInt(request.getParameter("feet"));
            flag3 = true;
        }catch(Exception e){
        }
    }
}
if (request.getParameter("inches") != null) {//set the number of inches in the
players height
    if (request.getParameter("inches") != ""){
        try{
            inches = Integer.parseInt(request.getParameter("inches"));
            if(inches > 11){
                flag4 = false;// number not less than 11 inches
            }else{
                flag4 = true;
            }
        }catch(Exception e){
        }
    }
}

```

```

        }
    }
    if (request.getParameter("Weight") != null) {//set the weight in pounds
        if (request.getParameter("Weight") != ""){
            try{
                weight = Integer.parseInt(request.getParameter("Weight"));
                if(weight > 500){//weight cannot be greater than 500 pounds
                    flag5 = false;
                }else{
                    flag5 = true;
                }
            }catch(Exception e){
            }
        }
    }

    int flagsTrue = 0;
    for(int x=0; x<6;x++){//check through the flags
        if(flags[x] == true){//check flag 0 to 5
            flagsTrue++;//if flag true, increment flagsTrue count
        }
        if(flagsTrue>5){//if all the flags are true, then process
            if(randHW == true){//if random height and weight is chosen run
standard setter
                player.Background(rName, pName, eyes, skin, hair, age, randHW,
height, weight);//background setter method
                session.setAttribute("player2", player);//set the updated object as
a session variable
                response.sendRedirect("background.jsp");//redirect to the next page
            }
            if(flag3 == true && flag4 == true && flag5 == true){//if the feet,
inches and weight are entered
                height = feet + "' " + inches + '"';
//append the
feet and inches to put them with appropriate units in a string format
                player.Background(rName, pName, eyes, skin, hair,
age, randHW, height, weight);//normal setter
                session.setAttribute("player2", player);//set the
updated object as a session variable
            }
            response.sendRedirect("background.jsp");//redirect to the next page
        }
    }
%>
<div id="navcontainer"><!-- nav bar for the top of the page -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>

</head>
<body>
<h1>Bram's character creator</h1>
<h2>
    Name and Appearance

```

```

</h2>
<form><!-- form for inputting choices -->
<div id="Leftcolumn"><!-- left column -->
<b>character's name:</b><br>
<input type="text" name="cName" maxlength="17"><br><br><br><!--
Character name input parallel to the real name -->
    <div id = "age" >Age: <input type="text" name="age"
maxlength="4"></div><br><!-- textboxes for age, eyes, skin, hair in the left
column -->
    <div id = "eyes" >Eyes: <input type="text" name="eyes"
maxlength="17"></div><br>
    <div id = "skin" >Skin: <input type="text" name="skin"
maxlength="17"></div><br>
    <div id = "hair" >Hair: <input type="text" name="hair"
maxlength="17"></div><br>
</div>
<div id = "center" align="center">
    <input type="submit" id = "fixedbutton"><!-- submit form button -->
</div>
<div id="rightcolumn"><!-- right column -->
<b>your name:</b><br>
<input type="text" name="rName" maxlength="17"><br><br><br><!-- real
name input -->
    Height <input type="text" name="feet" id="feet" maxlength="1"> foot
<!-- height in feet and inches textboxes -->
    <input type="text" name="inches" id="inches" maxlength="2"> inches
<br>
    <input type="checkbox" name="rand" value = "1">random height and
weight<br><!-- random height and weight checkbox -->
    Weight <input type="text" name="Weight" id="Weight" maxlength="3">
pounds<br> <!-- weight in pounds -->
</div>

</form></body>
</html>

```

```

background.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!-lets the user choose their background so the appropriate coursework setter
method is applied to their object-->
<!DOCTYPE html>
<html>
<head>
<script src="backgroundJ.js"></script><!-- import all relevant information loaders
-->
<script src="valid.js"></script><!-- import checkbox validator -->
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<%
    session.getAttribute("player2");//get up to date version
    coursework player = (coursework)session.getAttribute("player2");//create a
local version of the object for setting more values
    int selectedItem = 0;//initialised choice variables
    String choice = "";
    String choice2 = "";
    if (request.getParameter("background") != null) {
        selectedItem =
Integer.parseInt(request.getParameter("background"));//get chosen background
    }

    switch(selectedItem){//switch with chosen background
    case 0://acolyte
        if (request.getParameter("langH") != null) {//acolytes choose two
languages
            try{//catch errors, e.g. too many or few selections
                String[] a = request.getParameterValues("langH");//get
languages as an array
                choice = a[0];//set choices as each value in the array
                choice2 = a[1];
                player.Acolyte(choice, choice2);//set the acolyte
values with the language parameters
                session.setAttribute("player3", player);//set current
object as a session variable
                response.sendRedirect("acolyte.jsp");//redirect the
user to the acolyte page for more background specific choices
            }
            break;
        }catch(Exception e){
    }
}
case 1://charlatan
    if (request.getParameter("scam") != null) {//choose scam tools
        choice = request.getParameter("scam");
        player.Charlatan(choice);//set charlatan
        session.setAttribute("player3", player);//set session object
        response.sendRedirect("charlatan.jsp");//redirect
    }
    break;
case 2://criminal

```

```

        if (request.getParameter("game") != null) {//choose proficiency in a
game set
            choice = request.getParameter("game");
            player.Criminal(choice, false); //criminal setter, false as it
isn't the variant background
            session.setAttribute("player3", player); //set session object
            response.sendRedirect("criminal.jsp"); //redirect
        }
        break;
    case 3://entertainer
        if (request.getParameter("inst") != null) {
            choice = request.getParameter("inst"); //get chosen instrument
            player.Entertainer(choice, false); //setter, not variant so
false parameter
            session.setAttribute("player3", player); //set session object
            response.sendRedirect("entertainer.jsp"); //redirect
        }
        break;
    case 4://folk hero
        if (request.getParameter("artisan") != null) {
            choice = request.getParameter("artisan"); //get artisan tools
            player.FolkHero(choice); //set folk hero with parameters
            session.setAttribute("player3", player);
            response.sendRedirect("folkHero.jsp"); //redirect
        }
        break;
    case 5://guild artisan
        if (request.getParameter("artisan") != null) {
            choice = request.getParameter("artisan"); //get artisan tools
            if (request.getParameter("langH") != null) {
                choice2 = request.getParameter("langH"); //get chosen
language
                player.GuildArtisan(choice, choice2, false); //setter
with parameters, not variant so false
                session.setAttribute("player3", player);
                response.sendRedirect("guildArtisan.jsp"); //redirect
            }
        }
        break;
    case 6://hermit
        if (request.getParameter("langH") != null) {
            choice = request.getParameter("langH"); //get chosen language
            player.Hermit(choice); //set as hermit with language as
parameter
            session.setAttribute("player3", player);
            response.sendRedirect("hermit.jsp"); //redirect
        }
        break;
    case 7://noble
        if (request.getParameter("game") != null) {
            choice = request.getParameter("game"); //get game set
            if (request.getParameter("langH") != null) {
                choice2 = request.getParameter("langH"); //get chosen
language
                player.Noble(choice, choice2, false); //setter with
parameters and false as it isn't the variant
                session.setAttribute("player3", player);
                response.sendRedirect("noble.jsp"); //redirect
            }
        }
        break;
}

```

```

        }
    }
    break;
case 8://outlander
    if (request.getParameter("inst") != null) {
        choice = request.getParameter("inst");//get chosen instrument
        if (request.getParameter("langH") != null) {
            choice2 = request.getParameter("langH");//get chosen
language
parameters
player.Outlander(choice, choice2);//setter with
session.setAttribute("player3", player);
response.sendRedirect("outlander.jsp");//redirect
    }
}
break;
case 9://sage
if (request.getParameter("langH") != null) {
try{
String[] a = request.getParameterValues("langH");//get
two chosen languages
choice = a[0];
choice2 = a[1];
player.Sage(choice, choice2);//setter with two
parameters
session.setAttribute("player3", player);
response.sendRedirect("sage.jsp");//redirect
}catch(Exception e){
}
}
break;
case 10://sailor
player.Sailor(false);//setter for sailor, not variant
session.setAttribute("player3", player);
response.sendRedirect("sailor.jsp");//redirect
break;
case 11://soldier
if (request.getParameter("game") != null) {
choice = request.getParameter("game");//get chosen game set
player.Soldier(choice);//setter with parameter
session.setAttribute("player3", player);
response.sendRedirect("soldier.jsp");//redirect
}
break;
case 12://urchin
player.Urchin();//setter
session.setAttribute("player3", player);
response.sendRedirect("urchin.jsp");//redirect
break;
//variants
case 21://criminal variant, spy
if (request.getParameter("game") != null) {
choice = request.getParameter("game");//get chosen game set
player.Criminal(choice, true);//setter with game parameter and
variant true
session.setAttribute("player3", player);
response.sendRedirect("criminal.jsp");//redirect
}

```

```

        break;
    case 22://entertainer variant, gladiator
        if (request.getParameter("inst") != null) {
            choice = request.getParameter("inst");//get chosen instrument
            player.Entertainer(choice, true);//setter with parameter and
variant true
            session.setAttribute("player3", player);
            response.sendRedirect("entertainer.jsp");//redirect
        }
        break;
    case 23://guild artisan variant, guild merchant
        if (request.getParameter("artisan") != null) {
            choice = request.getParameter("artisan");//get artisan tools
            if (request.getParameter("langH") != null) {
                choice2 = request.getParameter("langH");//get chosen
language
                player.GuildArtisan(choice, choice2, true);//setter
with parameters and variant set to true
                session.setAttribute("player3", player);
                response.sendRedirect("guildArtisan.jsp");//redirect
            }
        }
        break;
    case 24://noble variant, knight
        if (request.getParameter("game") != null) {
            choice = request.getParameter("game");//chosen game set
            if (request.getParameter("langH") != null) {
                choice2 = request.getParameter("langH");//chosen
language
                player.Noble(choice, choice2, true);//setter with
variant set to true
                session.setAttribute("player3", player);
                response.sendRedirect("noble.jsp");//redirect
            }
        }
        break;
    case 25://sailor variant pirate
        player.Sailor(true);//setter, variant true
        session.setAttribute("player3", player);
        response.sendRedirect("sailor.jsp");//redirect
        break;
    }

%>
<div id="navcontainer"><!-- top of page nav bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Background
</h2>
<form><!-- form for background choice -->
    <div align = "center"><!-- centralised to keep to the layout -
->

```

```

        <input type="radio" name="background" id="Acolyte" value="0"
onclick = "acol();checkboxlimit(langH, 2)">Acolyte <br>
        <!-- background radio buttons, with imported methods from
backgroundJ for generating radio and checkbox buttons -->
        <!-- imported valid js page with validation for limiting
checkboxes to two choices -->
        <input type="radio" name="background" id="Charlatan" value="1"
onclick = "char();">Charlatan <br>
        <input type="radio" name="background" id="Criminal" value="2"
onclick = "crim();">Criminal
        <input type="radio" name="background" id="spy" value="21"
onclick = "crim();">Spy <br>
        <!-- variants are on the same line as their regular
counterparts -->
        <input type="radio" name="background" id="Entertainer"
value="3" onclick = "ent();">Entertainer
        <input type="radio" name="background" id="Gladiator"
value="22" onclick = "ent();">Gladiator <br>
        <input type="radio" name="background" id="FolkHero" value="4"
onclick = "folk();">Folk Hero <br>
        <input type="radio" name="background" id="GuildArisan"
value="5" onclick = "guild();">Guild Arisan
        <input type="radio" name="background" id="GuildMerchant"
value="23" onclick = "guild();">Guild Merchant<br>
        <input type="radio" name="background" id="Hermit" value="6"
onclick = "herm();">Hermit <br>
        <input type="radio" name="background" id="Noble" value="7"
onclick = "nob();">Noble
        <input type="radio" name="background" id="Knight" value="24"
onclick = "nob();knight();">Knight<br>
        <input type="radio" name="background" id="Outlander" value="8"
onclick = "outlander();">Outlander <br>
        <input type="radio" name="background" id="Sage" value="9"
onclick = "sage();">Sage <br>
        <input type="radio" name="background" id="Sailor" value="10"
onclick = "sailor();">Sailor
        <input type="radio" name="background" id="Pirate" value="25"
onclick = "sailor();pirate();">Pirate <br>
        <input type="radio" name="background" id="Soldier" value="11"
onclick = "sold();">Soldier <br>
        <input type="radio" name="background" id="Urchin" value="12"
onclick = "urch();">Urchin <br><br>
        </div>
        <h3 align="center"> proficiencies: </h3><!-- divs for
backgroundJ to add info and options to -->
        <div id = "profBox" align="center"></div>
        <div id = "objProf" align="center"></div>
        <h3 align="center"> languages: </h3>
        <div align="center">current languages: <%= player.languages
%> </div><br><!-- tell the player what languages they know -->
        <!-- should help avoid players picking the same language twice
-->
        <div id = "Lang" align="center"></div>
        <div id = "objLang" align="center"></div>
        <h3 align="center"> equipment: </h3>
        <div id = "equipment" align="center"></div>
        <div id = "objEquip" align="center"></div>
        <h3 align="center"> feature: </h3>

```

```
<div id = "feature" align="center"></div>
<div align = "center">
<input type="submit" id = "fixedbutton"><!-- submi button -->
</div>
</form>
</body>
</html>
```

## backgroundJ.js

```
//the dynamic radio button functions for the background choices, for in depth  
//comments see radio.js, the hElf function  
  
function acol(){//create acolyte information in divs  
    document.getElementById("profBox").innerHTML = "";  
    document.getElementById("profBox").innerHTML = "You become proficient in  
religion and insight";  
    document.getElementById("objProf").innerHTML = "";  
    document.getElementById("lang").innerHTML = "";  
    document.getElementById("lang").innerHTML = "You learn two new languages";  
    document.getElementById("objLang").innerHTML = "";  
    var objDiv = document.getElementById("objLang");  
  
    var radioItem1 = document.createElement("input");//create radio buttons for  
each of the language choices  
    radioItem1.type = "checkbox";  
    radioItem1.name = "langH";  
    radioItem1.value = "Dwarvish";  
  
    var radioItem2 = document.createElement("input");  
    radioItem2.type = "checkbox";  
    radioItem2.name = "langH";  
    radioItem2.value = "Elvish";  
  
    var radioItem3 = document.createElement("input");  
    radioItem3.type = "checkbox";  
    radioItem3.name = "langH";  
    radioItem3.value = "Giant";  
  
    var radioItem4 = document.createElement("input");  
    radioItem4.type = "checkbox";  
    radioItem4.name = "langH";  
    radioItem4.value = "Gnomish";  
  
    var radioItem5 = document.createElement("input");  
    radioItem5.type = "checkbox";  
    radioItem5.name = "langH";  
    radioItem5.value = "Goblin";  
  
  
    var radioItem6 = document.createElement("input");  
    radioItem6.type = "checkbox";  
    radioItem6.name = "langH";  
    radioItem6.value = "Halfling";  
  
  
    var radioItem7 = document.createElement("input");  
    radioItem7.type = "checkbox";  
    radioItem7.name = "langH";  
    radioItem7.value = "Orc";  
  
    var radioItem8 = document.createElement("input");  
    radioItem8.type = "checkbox";  
    radioItem8.name = "langH";  
    radioItem8.value = "Abyssal";  
  
    var radioItem9 = document.createElement("input");
```

```

radioItem9.type = "checkbox";
radioItem9.name = "langH";
radioItem9.value = "Celestial";

var radioItem10 = document.createElement("input");
radioItem10.type = "checkbox";
radioItem10.name = "langH";
radioItem10.value = "Draconic";

var radioItem11 = document.createElement("input");
radioItem11.type = "checkbox";
radioItem11.name = "langH";
radioItem11.value = "Deep Speech";

var radioItem12 = document.createElement("input");
radioItem12.type = "checkbox";
radioItem12.name = "langH";
radioItem12.value = "Infernal";

var radioItem13 = document.createElement("input");
radioItem13.type = "checkbox";
radioItem13.name = "langH";
radioItem13.value = "Primordial";

var radioItem14 = document.createElement("input");
radioItem14.type = "checkbox";
radioItem14.name = "langH";
radioItem14.value = "Sylvan";

var radioItem15 = document.createElement("input");
radioItem15.type = "checkbox";
radioItem15.name = "langH";
radioItem15.value = "Undercommon";

var objTextNode1 = document.createTextNode("Dwarvish");//create text for each
radio button
var objTextNode2 = document.createTextNode("Elvish");
var objTextNode3 = document.createTextNode("Giant");
var objTextNode4 = document.createTextNode("Gnomish");
var objTextNode5 = document.createTextNode("Goblin");
var objTextNode6 = document.createTextNode("Halfling");
var objTextNode7 = document.createTextNode("Orc");
var objTextNode8 = document.createTextNode("Abyssal");
var objTextNode9 = document.createTextNode("Celestial");
var objTextNode10 = document.createTextNode("Draconic");
var objTextNode11 = document.createTextNode("Deep Speech");
var objTextNode12 = document.createTextNode("Infernal");
var objTextNode13 = document.createTextNode("Primordial");
var objTextNode14 = document.createTextNode("Sylvan");
var objTextNode15 = document.createTextNode("Undercommon");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

```

```
var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
```

```

objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
document.getElementById("equipment").innerHTML = "";
document.getElementById("equipment").innerHTML = "You recieve: A holy
symbol," +
        " a prayer book, 5 sticks of incense, vestments, common
clothes, 15 gold pieces";
document.getElementById("objEquip").innerHTML = "";
document.getElementById("feature").innerHTML = "";
document.getElementById("feature").innerHTML = "Shelter the faithful";
}

function char(){//charlatan info function
    document.getElementById("profBox").innerHTML = "";//add info to boxes
    document.getElementById("profBox").innerHTML = "You become proficient in
deception," +
        " sleight of hand and with disguise kits and forgery kits";
    document.getElementById("objProf").innerHTML = "";
    document.getElementById("lang").innerHTML = "";
    document.getElementById("lang").innerHTML = "You learn no new languages";
    document.getElementById("objLang").innerHTML = "";
    document.getElementById("equipment").innerHTML = "";
    document.getElementById("equipment").innerHTML = "You recieve: A set of
fine clothes," +
        " a disguise kit, tools of your con, and 15 gold pieces";
    document.getElementById("objEquip").innerHTML = "";
    var objDiv = document.getElementById("objEquip");
    var radioItem1 = document.createElement("input");//create scam tool radio
buttons
    radioItem1.type = "radio";
}

```

```

radioItem1.name = "scam";
radioItem1.value = "ten stoppered bottles of coloured liquid";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "scam";
radioItem2.value = "a set of weighted dice";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "scam";
radioItem3.value = "A deck of marked cards";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "scam";
radioItem4.value = "A signet ring of an imaginary duke";

var objTextNode1 = document.createTextNode("ten stoppered bottles of
coloured liquid");//text
var objTextNode2 = document.createTextNode("a set of weighted dice");
var objTextNode3 = document.createTextNode("A deck of marked cards");
var objTextNode4 = document.createTextNode("A signet ring of an imaginary
duke");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

objDiv.appendChild(objLabel);//append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
document.getElementById("feature").innerHTML = "";
document.getElementById("feature").innerHTML = "False identity";
}

function crim(){//criminal funciton
document.getElementById("profBox").innerHTML = "";//create infor

```

```

document.getElementById("profBox").innerHTML = "You become proficient in
stealth, deception," +
    " thieves tools and a gaming set of your choice";
document.getElementById("objProf").innerHTML = "";
var objDiv = document.getElementById("objProf");
var radioItem1 = document.createElement("input");//create game set radio
buttons
radioItem1.type = "radio";
radioItem1.name = "game";
radioItem1.value = "Dice set";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "game";
radioItem2.value = "Dragonchess set";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "game";
radioItem3.value = "Playing card set";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "game";
radioItem4.value = "Three-dragon ante";

var objTextNode1 = document.createTextNode("Dice set");//text
var objTextNode2 = document.createTextNode("Dragonchess set");
var objTextNode3 = document.createTextNode("Playing card set");
var objTextNode4 = document.createTextNode("Three-dragon ante");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

objDiv.appendChild(objLabel);//append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
document.getElementById("lang").innerHTML = "";

```

```

document.getElementById("lang").innerHTML = "You learn no new languages";
document.getElementById("objLang").innerHTML = "";
document.getElementById("equipment").innerHTML = "";
document.getElementById("equipment").innerHTML = "A crowbar, A set of dark
common clothing, " +
    "a hood, 15 gold pieces";
document.getElementById("objEquip").innerHTML = "";
document.getElementById("feature").innerHTML = "";
document.getElementById("feature").innerHTML = "criminal contract";
}

function ent(){//entertainer function
    document.getElementById("profBox").innerHTML = "";//create info
    document.getElementById("profBox").innerHTML = "You become proficient in
acrobatics " +
        "and performance and with disguise kits and a msuical
intrument:";
    document.getElementById("objProf").innerHTML = "";
    var objDiv = document.getElementById("objProf");
    var radioItem1 = document.createElement("input");//create instrument radio
buttons
    radioItem1.type = "radio";
    radioItem1.name = "inst";
    radioItem1.value = "Bagpipes";

    var radioItem2 = document.createElement("input");
    radioItem2.type = "radio";
    radioItem2.name = "inst";
    radioItem2.value = "Drum";

    var radioItem3 = document.createElement("input");
    radioItem3.type = "radio";
    radioItem3.name = "inst";
    radioItem3.value = "Dulcimer";

    var radioItem4 = document.createElement("input");
    radioItem4.type = "radio";
    radioItem4.name = "inst";
    radioItem4.value = "Flute";

    var radioItem5 = document.createElement("input");
    radioItem5.type = "radio";
    radioItem5.name = "inst";
    radioItem5.value = "Lute";

    var radioItem6 = document.createElement("input");
    radioItem6.type = "radio";
    radioItem6.name = "inst";
    radioItem6.value = "Lyre";

    var radioItem7 = document.createElement("input");
    radioItem7.type = "radio";
    radioItem7.name = "inst";
    radioItem7.value = "Horn";

    var radioItem8 = document.createElement("input");
    radioItem8.type = "radio";
}

```

```

radioItem8.name = "inst";
radioItem8.value = "Pan flute";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "inst";
radioItem9.value = "Shawm";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "inst";
radioItem10.value = "Viol";

var objTextNode1 = document.createTextNode("Bagpipes");//text
var objTextNode2 = document.createTextNode("Drum");
var objTextNode3 = document.createTextNode("Dulcimer");
var objTextNode4 = document.createTextNode("Flute");
var objTextNode5 = document.createTextNode("Lute");
var objTextNode6 = document.createTextNode("Lyre");
var objTextNode7 = document.createTextNode("Horn");
var objTextNode8 = document.createTextNode("Pan flute");
var objTextNode9 = document.createTextNode("Shawm");
var objTextNode10 = document.createTextNode("Viol");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

```

```

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
document.getElementById("lang").innerHTML = "";
document.getElementById("lang").innerHTML = "You learn no new languages";
document.getElementById("objLang").innerHTML = "";
document.getElementById("equipment").innerHTML = "";
document.getElementById("equipment").innerHTML = "You receive: Your musical instrument," +
                                              " the favor of an admirer, a costume and 15 gold pieces";
document.getElementById("objEquip").innerHTML = "";
document.getElementById("feature").innerHTML = "";
document.getElementById("feature").innerHTML = "By popular demand";
}

function folk(){//folk hero function
  document.getElementById("profBox").innerHTML = ""; //add info
  document.getElementById("profBox").innerHTML = "You become proficient in animal" +
                                              "handling, survival, vehicles(land), artisans tools of choice:";
  document.getElementById("objProf").innerHTML = "";
  var objDiv = document.getElementById("objProf");

  var radioItem1 = document.createElement("input"); //add artisan tool choice
  radioItem1.type = "radio";
  radioItem1.name = "artisan";
  radioItem1.value = "Alchemist's supplies";
}

```

```

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "artisan";
radioItem2.value = "Brewer's supplies";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "artisan";
radioItem3.value = "Calligrapher's supplies";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "artisan";
radioItem4.value = "Carpenter's tools";

var radioItem5 = document.createElement("input");
radioItem5.type = "radio";
radioItem5.name = "artisan";
radioItem5.value = "Cobbler's tools";

var radioItem6 = document.createElement("input");
radioItem6.type = "radio";
radioItem6.name = "artisan";
radioItem6.value = "Cartographer's tools";

var radioItem7 = document.createElement("input");
radioItem7.type = "radio";
radioItem7.name = "artisan";
radioItem7.value = "Cook's utensils";

var radioItem8 = document.createElement("input");
radioItem8.type = "radio";
radioItem8.name = "artisan";
radioItem8.value = "Glassblower's tools";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "artisan";
radioItem9.value = "Jeweler's tools";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "artisan";
radioItem10.value = "Leatherworker's tools";

var radioItem11 = document.createElement("input");
radioItem11.type = "radio";
radioItem11.name = "artisan";
radioItem11.value = "Mason's tools";

var radioItem12 = document.createElement("input");
radioItem12.type = "radio";
radioItem12.name = "artisan";
radioItem12.value = "Painter's supplies";

var radioItem13 = document.createElement("input");
radioItem13.type = "radio";
radioItem13.name = "artisan";

```

```

radioItem13.value = "Potter's tools";

var radioItem14 = document.createElement("input");
radioItem14.type = "radio";
radioItem14.name = "artisan";
radioItem14.value = "Smith's tools";

var radioItem15 = document.createElement("input");
radioItem15.type = "radio";
radioItem15.name = "artisan";
radioItem15.value = "Tinker's tools";

var radioItem16 = document.createElement("input");
radioItem16.type = "radio";
radioItem16.name = "artisan";
radioItem16.value = "Weaver's tools";

var radioItem17 = document.createElement("input");
radioItem17.type = "radio";
radioItem17.name = "artisan";
radioItem17.value = "Woodcarver's tools";

var objTextNode1 = document.createTextNode("Alchemist's supplies");//text
var objTextNode2 = document.createTextNode("Brewer's supplies");
var objTextNode3 = document.createTextNode("Calligrapher's supplies");
var objTextNode4 = document.createTextNode("Carpenter's tools");
var objTextNode5 = document.createTextNode("Cartographer's tools");
var objTextNode6 = document.createTextNode("Cobbler's tools");
var objTextNode7 = document.createTextNode("Cook's utensils");
var objTextNode8 = document.createTextNode("Glassblower's tools");
var objTextNode9 = document.createTextNode("Jeweler's tools");
var objTextNode10 = document.createTextNode("Leatherworker's tools ");
var objTextNode11 = document.createTextNode("Mason's tools");
var objTextNode12 = document.createTextNode("Painter's supplies");
var objTextNode13 = document.createTextNode("Potter's tools");
var objTextNode14 = document.createTextNode("Smith's tools");
var objTextNode15 = document.createTextNode("Tinker's tools");
var objTextNode16 = document.createTextNode("Weaver's tools");
var objTextNode17 = document.createTextNode("Woodcarver's tools");

    var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);

```

```
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

var objLabel16 = document.createElement("label");
```

```

objLabel16.htmlFor = radioItem16.id;
objLabel16.appendChild(radioItem16);
objLabel16.appendChild(objTextNode16);

var objLabel17 = document.createElement("label");
objLabel17.htmlFor = radioItem17.id;
objLabel17.appendChild(radioItem17);
objLabel17.appendChild(objTextNode17);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.innerHTML += "<br>"; //break line
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
objDiv.appendChild(objLabel16);
objDiv.appendChild(objLabel17);
document.getElementById("lang").innerHTML = "";
document.getElementById("lang").innerHTML = "You learn no new languages";
document.getElementById("objLang").innerHTML = "";
document.getElementById("equipment").innerHTML = "";
document.getElementById("equipment").innerHTML = "You receive: your
proficient artisan tools, a shovel, an iron pot, a set of common clothes, 15 gold
pieces";
document.getElementById("objEquip").innerHTML = "";
document.getElementById("feature").innerHTML = "";
document.getElementById("feature").innerHTML = "Rustic hospitality";
}

function guild(){//guild artisan function
document.getElementById("profBox").innerHTML = ""; //add info
document.getElementById("profBox").innerHTML = "You become proficient in
insight, persuasion, artisans tools of choice:";
document.getElementById("objProf").innerHTML = "";
var objDiv = document.getElementById("objProf");

var radioItem1 = document.createElement("input"); //add artisan tool radio
buttons
radioItem1.type = "radio";
radioItem1.name = "artisan";
radioItem1.value = "Alchemist's supplies";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "artisan";
radioItem2.value = "Brewer's supplies";

var radioItem3 = document.createElement("input");

```

```
radioItem3.type = "radio";
radioItem3.name = "artisan";
radioItem3.value = "Calligrapher's supplies";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "artisan";
radioItem4.value = "Carpenter's tools";

var radioItem5 = document.createElement("input");
radioItem5.type = "radio";
radioItem5.name = "artisan";
radioItem5.value = "Cobbler's tools";

var radioItem6 = document.createElement("input");
radioItem6.type = "radio";
radioItem6.name = "artisan";
radioItem6.value = "Cartographer's tools";

var radioItem7 = document.createElement("input");
radioItem7.type = "radio";
radioItem7.name = "artisan";
radioItem7.value = "Cook's utensils";

var radioItem8 = document.createElement("input");
radioItem8.type = "radio";
radioItem8.name = "artisan";
radioItem8.value = "Glassblower's tools";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "artisan";
radioItem9.value = "Jeweler's tools";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "artisan";
radioItem10.value = "Leatherworker's tools";

var radioItem11 = document.createElement("input");
radioItem11.type = "radio";
radioItem11.name = "artisan";
radioItem11.value = "Mason's tools";

var radioItem12 = document.createElement("input");
radioItem12.type = "radio";
radioItem12.name = "artisan";
radioItem12.value = "Painter's supplies";

var radioItem13 = document.createElement("input");
radioItem13.type = "radio";
radioItem13.name = "artisan";
radioItem13.value = "Potter's tools";

var radioItem14 = document.createElement("input");
radioItem14.type = "radio";
radioItem14.name = "artisan";
radioItem14.value = "Smith's tools";
```

```

var radioItem15 = document.createElement("input");
radioItem15.type = "radio";
radioItem15.name = "artisan";
radioItem15.value = "Tinker's tools";

var radioItem16 = document.createElement("input");
radioItem16.type = "radio";
radioItem16.name = "artisan";
radioItem16.value = "Weaver's tools";

var radioItem17 = document.createElement("input");
radioItem17.type = "radio";
radioItem17.name = "artisan";
radioItem17.value = "Woodcarver's tools";

var objTextNode1 = document.createTextNode("Alchemist's supplies");//text
var objTextNode2 = document.createTextNode("Brewer's supplies");
var objTextNode3 = document.createTextNode("Calligrapher's supplies");
var objTextNode4 = document.createTextNode("Carpenter's tools");
var objTextNode5 = document.createTextNode("Cartographer's tools");
var objTextNode6 = document.createTextNode("Cobbler's tools");
var objTextNode7 = document.createTextNode("Cook's utensils");
var objTextNode8 = document.createTextNode("Glassblower's tools");
var objTextNode9 = document.createTextNode("Jeweler's tools");
var objTextNode10 = document.createTextNode("Leatherworker's tools");
var objTextNode11 = document.createTextNode("Mason's tools");
var objTextNode12 = document.createTextNode("Painter's supplies");
var objTextNode13 = document.createTextNode("Potter's tools");
var objTextNode14 = document.createTextNode("Smith's tools");
var objTextNode15 = document.createTextNode("Tinker's tools");
var objTextNode16 = document.createTextNode("Weaver's tools");
var objTextNode17 = document.createTextNode("Woodcarver's tools");

    var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

```

```

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

var objLabel16 = document.createElement("label");
objLabel16.htmlFor = radioItem16.id;
objLabel16.appendChild(radioItem16);
objLabel16.appendChild(objTextNode16);

var objLabel17 = document.createElement("label");
objLabel17.htmlFor = radioItem17.id;

```

```

objLabel17.appendChild(radioItem17);
objLabel17.appendChild(objTextNode17);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.innerHTML += "<br>"; //break line
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
objDiv.appendChild(objLabel16);
objDiv.appendChild(objLabel17);
document.getElementById("lang").innerHTML = "";
document.getElementById("lang").innerHTML = "You learn one new language";
document.getElementById("objLang").innerHTML = "";
var objDiv = document.getElementById("objLang");

var radioItem1 = document.createElement("input"); //add language radio buttons
radioItem1.type = "radio";
radioItem1.name = "langH";
radioItem1.value = "Dwarvish";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "langH";
radioItem2.value = "Elvish";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "langH";
radioItem3.value = "Giant";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "langH";
radioItem4.value = "Gnomish";

var radioItem5 = document.createElement("input");
radioItem5.type = "radio";
radioItem5.name = "langH";
radioItem5.value = "Goblin";

var radioItem6 = document.createElement("input");
radioItem6.type = "radio";
radioItem6.name = "langH";
radioItem6.value = "Halfling";

```

```

var radioItem7 = document.createElement("input");
radioItem7.type = "radio";
radioItem7.name = "langH";
radioItem7.value = "Orc";

var radioItem8 = document.createElement("input");
radioItem8.type = "radio";
radioItem8.name = "langH";
radioItem8.value = "Abyssal";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "langH";
radioItem9.value = "Celestial";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "langH";
radioItem10.value = "Draconic";

var radioItem11 = document.createElement("input");
radioItem11.type = "radio";
radioItem11.name = "langH";
radioItem11.value = "Deep Speech";

var radioItem12 = document.createElement("input");
radioItem12.type = "radio";
radioItem12.name = "langH";
radioItem12.value = "Infernal";

var radioItem13 = document.createElement("input");
radioItem13.type = "radio";
radioItem13.name = "langH";
radioItem13.value = "Primordial";

var radioItem14 = document.createElement("input");
radioItem14.type = "radio";
radioItem14.name = "langH";
radioItem14.value = "Sylvan";

var radioItem15 = document.createElement("input");
radioItem15.type = "radio";
radioItem15.name = "langH";
radioItem15.value = "Undercommon";

var objTextNode1 = document.createTextNode("Dwarvish");//text
var objTextNode2 = document.createTextNode("Elvish");
var objTextNode3 = document.createTextNode("Giant");
var objTextNode4 = document.createTextNode("Gnomish");
var objTextNode5 = document.createTextNode("Goblin");
var objTextNode6 = document.createTextNode("Halfling");
var objTextNode7 = document.createTextNode("Orc");
var objTextNode8 = document.createTextNode("Abyssal");
var objTextNode9 = document.createTextNode("Celestial");
var objTextNode10 = document.createTextNode("Draconic");
var objTextNode11 = document.createTextNode("Deep Speech");
var objTextNode12 = document.createTextNode("Infernal");

```

```

var objTextNode13 = document.createTextNode("Primordial");
var objTextNode14 = document.createTextNode("Sylvan");
var objTextNode15 = document.createTextNode("Undercommon");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");

```

```

objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
objDiv.appendChild(objLabel16);
objDiv.appendChild(objLabel17);
objDiv.appendChild(objLabel18);
objDiv.appendChild(objLabel19);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);

document.getElementById("equipment").innerHTML = "";
document.getElementById("equipment").innerHTML = "You receive: your
proficient artisan tools," +
    " A letter of introduction from your guild, a set of
traveller's clothes, 15 gold pieces";
document.getElementById("objEquip").innerHTML = "";
document.getElementById("feature").innerHTML = "";
document.getElementById("feature").innerHTML = "Rustic hospitality";
}

function herm(){//hermit function
    document.getElementById("profBox").innerHTML = "";//add info
    document.getElementById("profBox").innerHTML = "You become proficient in
medicine, religion and with Herbalism kits";
    document.getElementById("objProf").innerHTML = "";
    document.getElementById("lang").innerHTML = "";
    document.getElementById("lang").innerHTML = "You learn a new language";
    document.getElementById("objLang").innerHTML = "";
    var objDiv = document.getElementById("objLang");
}

```

```

var radioItem1 = document.createElement("input");//language radio buttons
radioItem1.type = "radio";
radioItem1.name = "langH";
radioItem1.value = "Dwarvish";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "langH";
radioItem2.value = "Elvish";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "langH";
radioItem3.value = "Giant";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "langH";
radioItem4.value = "Gnomish";

var radioItem5 = document.createElement("input");
radioItem5.type = "radio";
radioItem5.name = "langH";
radioItem5.value = "Goblin";

var radioItem6 = document.createElement("input");
radioItem6.type = "radio";
radioItem6.name = "langH";
radioItem6.value = "Halfling";

var radioItem7 = document.createElement("input");
radioItem7.type = "radio";
radioItem7.name = "langH";
radioItem7.value = "Orc";

var radioItem8 = document.createElement("input");
radioItem8.type = "radio";
radioItem8.name = "langH";
radioItem8.value = "Abyssal";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "langH";
radioItem9.value = "Celestial";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "langH";
radioItem10.value = "Draconic";

var radioItem11 = document.createElement("input");
radioItem11.type = "radio";
radioItem11.name = "langH";
radioItem11.value = "Deep Speech";

```

```

var radioItem12 = document.createElement("input");
radioItem12.type = "radio";
radioItem12.name = "langH";
radioItem12.value = "Infernal";

var radioItem13 = document.createElement("input");
radioItem13.type = "radio";
radioItem13.name = "langH";
radioItem13.value = "Primordial";

var radioItem14 = document.createElement("input");
radioItem14.type = "radio";
radioItem14.name = "langH";
radioItem14.value = "Sylvan";

var radioItem15 = document.createElement("input");
radioItem15.type = "radio";
radioItem15.name = "langH";
radioItem15.value = "Undercommon";

var objTextNode1 = document.createTextNode("Dwarvish");//add text
var objTextNode2 = document.createTextNode("Elvish");
var objTextNode3 = document.createTextNode("Giant");
var objTextNode4 = document.createTextNode("Gnomish");
var objTextNode5 = document.createTextNode("Goblin");
var objTextNode6 = document.createTextNode("Halfling");
var objTextNode7 = document.createTextNode("Orc");
var objTextNode8 = document.createTextNode("Abyssal");
var objTextNode9 = document.createTextNode("Celestial");
var objTextNode10 = document.createTextNode("Draconic");
var objTextNode11 = document.createTextNode("Deep Speech");
var objTextNode12 = document.createTextNode("Infernal");
var objTextNode13 = document.createTextNode("Primordial");
var objTextNode14 = document.createTextNode("Sylvan");
var objTextNode15 = document.createTextNode("Undercommon");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

```

```
var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);
```

```

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
    document.getElementById("equipment").innerHTML = "";
    document.getElementById("equipment").innerHTML = "You receive: A scroll
case stuffed full of notes from your studies or prayers, a winter blanket, a set
of common clothes, a herbalism kit, 5 gold pieces";
    document.getElementById("objEquip").innerHTML = "";
    document.getElementById("feature").innerHTML = "";
    document.getElementById("feature").innerHTML = "Discovery";
}
function nob(){//noble function
    document.getElementById("profBox").innerHTML = "";//add info
    document.getElementById("profBox").innerHTML = "You become proficient in
history, persuasion and a gaming set of your choice";
    document.getElementById("objProf").innerHTML = "";
    var objDiv = document.getElementById("objProf");
    var radioItem1 = document.createElement("input");//game set radio buttons
    radioItem1.type = "radio";
    radioItem1.name = "game";
    radioItem1.value = "Dice set";

    var radioItem2 = document.createElement("input");
    radioItem2.type = "radio";
    radioItem2.name = "game";
    radioItem2.value = "Dragonchess set";

    var radioItem3 = document.createElement("input");
    radioItem3.type = "radio";
    radioItem3.name = "game";
    radioItem3.value = "Playing card set";

    var radioItem4 = document.createElement("input");
    radioItem4.type = "radio";
    radioItem4.name = "game";
    radioItem4.value = "Three-dragon ante";

    var objTextNode1 = document.createTextNode("Dice set");//text
    var objTextNode2 = document.createTextNode("Dragonchess set");
    var objTextNode3 = document.createTextNode("Playing card set");
    var objTextNode4 = document.createTextNode("Three-dragon ante");

    var objLabel = document.createElement("label");
    objLabel.htmlFor = radioItem1.id;
}

```

```

objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

objDiv.appendChild(objLabel);//append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
document.getElementById("lang").innerHTML = "";
document.getElementById("lang").innerHTML = "You learn a new language";
document.getElementById("objLang").innerHTML = "";
var objDiv = document.getElementById("objLang");

var radioItem1 = document.createElement("input");//language radio buttons
radioItem1.type = "radio";
radioItem1.name = "langH";
radioItem1.value = "Dwarvish";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "langH";
radioItem2.value = "Elvish";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "langH";
radioItem3.value = "Giant";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "langH";
radioItem4.value = "Gnomish";

var radioItem5 = document.createElement("input");
radioItem5.type = "radio";
radioItem5.name = "langH";
radioItem5.value = "Goblin";

var radioItem6 = document.createElement("input");
radioItem6.type = "radio";
radioItem6.name = "langH";
radioItem6.value = "Halfling";

```

```

var radioItem7 = document.createElement("input");
radioItem7.type = "radio";
radioItem7.name = "langH";
radioItem7.value = "Orc";

var radioItem8 = document.createElement("input");
radioItem8.type = "radio";
radioItem8.name = "langH";
radioItem8.value = "Abyssal";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "langH";
radioItem9.value = "Celestial";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "langH";
radioItem10.value = "Draconic";

var radioItem11 = document.createElement("input");
radioItem11.type = "radio";
radioItem11.name = "langH";
radioItem11.value = "Deep Speech";

var radioItem12 = document.createElement("input");
radioItem12.type = "radio";
radioItem12.name = "langH";
radioItem12.value = "Infernal";

var radioItem13 = document.createElement("input");
radioItem13.type = "radio";
radioItem13.name = "langH";
radioItem13.value = "Primordial";

var radioItem14 = document.createElement("input");
radioItem14.type = "radio";
radioItem14.name = "langH";
radioItem14.value = "Sylvan";

var radioItem15 = document.createElement("input");
radioItem15.type = "radio";
radioItem15.name = "langH";
radioItem15.value = "Undercommon";

var objTextNode1 = document.createTextNode("Dwarvish");//text
var objTextNode2 = document.createTextNode("Elvish");
var objTextNode3 = document.createTextNode("Giant");
var objTextNode4 = document.createTextNode("Gnomish");
var objTextNode5 = document.createTextNode("Goblin");
var objTextNode6 = document.createTextNode("Halfling");
var objTextNode7 = document.createTextNode("Orc");
var objTextNode8 = document.createTextNode("Abyssal");
var objTextNode9 = document.createTextNode("Celestial");
var objTextNode10 = document.createTextNode("Draconic");
var objTextNode11 = document.createTextNode("Deep Speech");

```

```

var objTextNode12 = document.createTextNode("Infernal");
var objTextNode13 = document.createTextNode("Primordial");
var objTextNode14 = document.createTextNode("Sylvan");
var objTextNode15 = document.createTextNode("Undercommon");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

```

```

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
document.getElementById("equipment").innerHTML = "";
document.getElementById("equipment").innerHTML = "A set of fine clothes, a
signet ring, " +
            "a scroll of pedigree, 25 gold pieces";
document.getElementById("objEquip").innerHTML = "";
document.getElementById("feature").innerHTML = "";
document.getElementById("feature").innerHTML = "Position of privilege";

}

function knight(){//knight function
    document.getElementById("feature").innerHTML = "";//variant feat info
    document.getElementById("feature").innerHTML = "Retainers";
}
function outlander(){//outlander function
    document.getElementById("profBox").innerHTML = "";//add text
    document.getElementById("profBox").innerHTML = "You become proficient in
athletics, " +

```

```

        " survival and a musical instrument of choice:";
document.getElementById("objProf").innerHTML = "";
var objDiv = document.getElementById("objProf");
var radioItem1 = document.createElement("input");//instrument radio buttons
radioItem1.type = "radio";
radioItem1.name = "inst";
radioItem1.value = "Bagpipes";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "inst";
radioItem2.value = "Drum";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "inst";
radioItem3.value = "Dulcimer";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "inst";
radioItem4.value = "Flute";

var radioItem5 = document.createElement("input");
radioItem5.type = "radio";
radioItem5.name = "inst";
radioItem5.value = "Lute";

var radioItem6 = document.createElement("input");
radioItem6.type = "radio";
radioItem6.name = "inst";
radioItem6.value = "Lyre";

var radioItem7 = document.createElement("input");
radioItem7.type = "radio";
radioItem7.name = "inst";
radioItem7.value = "Horn";

var radioItem8 = document.createElement("input");
radioItem8.type = "radio";
radioItem8.name = "inst";
radioItem8.value = "Pan flute";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "inst";
radioItem9.value = "Shawm";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "inst";
radioItem10.value = "Viol";

var objTextNode1 = document.createTextNode("Bagpipes");//text
var objTextNode2 = document.createTextNode("Drum");
var objTextNode3 = document.createTextNode("Dulcimer");

```

```

var objTextNode4 = document.createTextNode("Flute");
var objTextNode5 = document.createTextNode("Lute");
var objTextNode6 = document.createTextNode("Lyre");
var objTextNode7 = document.createTextNode("Horn");
var objTextNode8 = document.createTextNode("Pan flute");
var objTextNode9 = document.createTextNode("Shawm");
var objTextNode10 = document.createTextNode("Viol");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");

```

```

objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
document.getElementById("lang").innerHTML = "";
document.getElementById("lang").innerHTML = "You learn a new language";
document.getElementById("objLang").innerHTML = "";
var objDiv = document.getElementById("objLang"); //add language

var radioItem1 = document.createElement("input");
radioItem1.type = "radio";
radioItem1.name = "langH";
radioItem1.value = "Dwarvish";

var radioItem2 = document.createElement("input");
radioItem2.type = "radio";
radioItem2.name = "langH";
radioItem2.value = "Elvish";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "langH";
radioItem3.value = "Giant";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "langH";
radioItem4.value = "Gnomish";

var radioItem5 = document.createElement("input");
radioItem5.type = "radio";
radioItem5.name = "langH";
radioItem5.value = "Goblin";

var radioItem6 = document.createElement("input");
radioItem6.type = "radio";
radioItem6.name = "langH";
radioItem6.value = "Halfling";

var radioItem7 = document.createElement("input");
radioItem7.type = "radio";
radioItem7.name = "langH";
radioItem7.value = "Orc";

var radioItem8 = document.createElement("input");

```

```

radioItem8.type = "radio";
radioItem8.name = "langH";
radioItem8.value = "Abyssal";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "langH";
radioItem9.value = "Celestial";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "langH";
radioItem10.value = "Draconic";

var radioItem11 = document.createElement("input");
radioItem11.type = "radio";
radioItem11.name = "langH";
radioItem11.value = "Deep Speech";

var radioItem12 = document.createElement("input");
radioItem12.type = "radio";
radioItem12.name = "langH";
radioItem12.value = "Infernal";

var radioItem13 = document.createElement("input");
radioItem13.type = "radio";
radioItem13.name = "langH";
radioItem13.value = "Primordial";

var radioItem14 = document.createElement("input");
radioItem14.type = "radio";
radioItem14.name = "langH";
radioItem14.value = "Sylvan";

var radioItem15 = document.createElement("input");
radioItem15.type = "radio";
radioItem15.name = "langH";
radioItem15.value = "Undercommon";

var objTextNode1 = document.createTextNode("Dwarvish");//text
var objTextNode2 = document.createTextNode("Elvish");
var objTextNode3 = document.createTextNode("Giant");
var objTextNode4 = document.createTextNode("Gnomish");
var objTextNode5 = document.createTextNode("Goblin");
var objTextNode6 = document.createTextNode("Halfling");
var objTextNode7 = document.createTextNode("Orc");
var objTextNode8 = document.createTextNode("Abyssal");
var objTextNode9 = document.createTextNode("Celestial");
var objTextNode10 = document.createTextNode("Draconic");
var objTextNode11 = document.createTextNode("Deep Speech");
var objTextNode12 = document.createTextNode("Infernal");
var objTextNode13 = document.createTextNode("Primordial");
var objTextNode14 = document.createTextNode("Sylvan");
var objTextNode15 = document.createTextNode("Undercommon");

var objLabel = document.createElement("label");

```

```

objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;

```

```

objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
document.getElementById("equipment").innerHTML = "";
document.getElementById("equipment").innerHTML = "You receive: A staff, a
hunting trap," +
    " a trophy from an animal you killed, a set of traveller's
clothes, 10 gold pieces";
document.getElementById("objEquip").innerHTML = "";
document.getElementById("feature").innerHTML = "";
document.getElementById("feature").innerHTML = "Wanderer";

}

function sage(){//sage function
document.getElementById("profBox").innerHTML = ""; //add info
document.getElementById("profBox").innerHTML = "You become proficient in
Arcana and History";
document.getElementById("objProf").innerHTML = "";
document.getElementById("lang").innerHTML = "";
document.getElementById("lang").innerHTML = "You learn two new languages";
document.getElementById("objLang").innerHTML = "";
var objDiv = document.getElementById("objLang");

var radioItem1 = document.createElement("input"); //add language radio
buttons
radioItem1.type = "checkbox";

```

```

radioItem1.name = "langH";
radioItem1.value = "Dwarvish";

var radioItem2 = document.createElement("input");
radioItem2.type = "checkbox";
radioItem2.name = "langH";
radioItem2.value = "Elvish";

var radioItem3 = document.createElement("input");
radioItem3.type = "checkbox";
radioItem3.name = "langH";
radioItem3.value = "Giant";

var radioItem4 = document.createElement("input");
radioItem4.type = "checkbox";
radioItem4.name = "langH";
radioItem4.value = "Gnomish";

var radioItem5 = document.createElement("input");
radioItem5.type = "checkbox";
radioItem5.name = "langH";
radioItem5.value = "Goblin";

var radioItem6 = document.createElement("input");
radioItem6.type = "checkbox";
radioItem6.name = "langH";
radioItem6.value = "Halfling";

var radioItem7 = document.createElement("input");
radioItem7.type = "checkbox";
radioItem7.name = "langH";
radioItem7.value = "Orc";

var radioItem8 = document.createElement("input");
radioItem8.type = "checkbox";
radioItem8.name = "langH";
radioItem8.value = "Abyssal";

var radioItem9 = document.createElement("input");
radioItem9.type = "checkbox";
radioItem9.name = "langH";
radioItem9.value = "Celestial";

var radioItem10 = document.createElement("input");
radioItem10.type = "checkbox";
radioItem10.name = "langH";
radioItem10.value = "Draconic";

var radioItem11 = document.createElement("input");
radioItem11.type = "checkbox";
radioItem11.name = "langH";
radioItem11.value = "Deep Speech";

var radioItem12 = document.createElement("input");
radioItem12.type = "checkbox";
radioItem12.name = "langH";

```

```

radioItem12.value = "Infernal";

var radioItem13 = document.createElement("input");
radioItem13.type = "checkbox";
radioItem13.name = "langH";
radioItem13.value = "Primordial";

var radioItem14 = document.createElement("input");
radioItem14.type = "checkbox";
radioItem14.name = "langH";
radioItem14.value = "Sylvan";

var radioItem15 = document.createElement("input");
radioItem15.type = "checkbox";
radioItem15.name = "langH";
radioItem15.value = "Undercommon";

var objTextNode1 = document.createTextNode("Dwarvish");//text
var objTextNode2 = document.createTextNode("Elvish");
var objTextNode3 = document.createTextNode("Giant");
var objTextNode4 = document.createTextNode("Gnomish");
var objTextNode5 = document.createTextNode("Goblin");
var objTextNode6 = document.createTextNode("Halfling");
var objTextNode7 = document.createTextNode("Orc");
var objTextNode8 = document.createTextNode("Abyssal");
var objTextNode9 = document.createTextNode("Celestial");
var objTextNode10 = document.createTextNode("Draconic");
var objTextNode11 = document.createTextNode("Deep Speech");
var objTextNode12 = document.createTextNode("Infernal");
var objTextNode13 = document.createTextNode("Primordial");
var objTextNode14 = document.createTextNode("Sylvan");
var objTextNode15 = document.createTextNode("Undercommon");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);

```

```

objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);

```

```

objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);
    document.getElementById("equipment").innerHTML = "";
    document.getElementById("equipment").innerHTML = "You receive: A bottle of
black ink, a quill, a small knife, a letter" +
                    " from a dead colleague posing a question you have not been
able to answer, a set of common clothes, 10 gold pieces";
    document.getElementById("objEquip").innerHTML = "";
    document.getElementById("feature").innerHTML = "";
    document.getElementById("feature").innerHTML = "Researcher";
}
function sailor(){//sailor function
    document.getElementById("profBox").innerHTML = "";//add info
    document.getElementById("profBox").innerHTML = "You become proficient in
athletics, perception and" +
                    " with navigators tools and vehicles(water)";
    document.getElementById("objProf").innerHTML = "";
    document.getElementById("lang").innerHTML = "";
    document.getElementById("lang").innerHTML = "You learn no new languages";
    document.getElementById("objLang").innerHTML = "";
    document.getElementById("equipment").innerHTML = "";
    document.getElementById("equipment").innerHTML = "You receive: A belaying
pin(club), 50 feet" +
                    " of silk rope, a lucky charm, a set of common clothes, 10
gold pieces ";
    document.getElementById("objEquip").innerHTML = "";
    document.getElementById("feature").innerHTML = "";
    document.getElementById("feature").innerHTML = "Ship's passage";
}
function pirate(){//pirate function
    document.getElementById("feature").innerHTML = "";//replace feat info
    document.getElementById("feature").innerHTML = "Bad reputation";
}
function sold(){//soldier function
    document.getElementById("profBox").innerHTML = "";//add info
    document.getElementById("profBox").innerHTML = "You become proficient in
athletics, intimidation," +
                    " vehicles(land) and a gaming set of your choice";
    document.getElementById("objProf").innerHTML = "";
    var objDiv = document.getElementById("objProf");
    var radioItem1 = document.createElement("input");//game set radio buttons
    radioItem1.type = "radio";
    radioItem1.name = "game";
    radioItem1.value = "Dice set";

    var radioItem2 = document.createElement("input");
    radioItem2.type = "radio";
    radioItem2.name = "game";

```

```

radioItem2.value = "Dragonchess set";

var radioItem3 = document.createElement("input");
radioItem3.type = "radio";
radioItem3.name = "game";
radioItem3.value = "Playing card set";

var radioItem4 = document.createElement("input");
radioItem4.type = "radio";
radioItem4.name = "game";
radioItem4.value = "Three-dragon ante";

var objTextNode1 = document.createTextNode("Dice set");//text
var objTextNode2 = document.createTextNode("Dragonchess set");
var objTextNode3 = document.createTextNode("Playing card set");
var objTextNode4 = document.createTextNode("Three-dragon ante");

var objLabel1 = document.createElement("label");
objLabel1.htmlFor = radioItem1.id;
objLabel1.appendChild(radioItem1);
objLabel1.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

objDiv.appendChild(objLabel);//append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
document.getElementById("lang").innerHTML = "";
document.getElementById("lang").innerHTML = "You learn no new languages";
document.getElementById("objLang").innerHTML = "";
document.getElementById("equipment").innerHTML = "";
document.getElementById("equipment").innerHTML = "You recieve: An isnignia of your rank, a trophy" +
    " from a fallen enemy, a game set, a set of common clothes, 10 gold pieces";
document.getElementById("objEquip").innerHTML = "";
document.getElementById("feature").innerHTML = "";
document.getElementById("feature").innerHTML = "Military rank";
}

function urch(){//urchin function
    document.getElementById("profBox").innerHTML = "";//add info
}

```

```
document.getElementById("profBox").innerHTML = "You become proficient in  
sleight of hand, stealth and with" +  
    " disguise kits and thieves tools";  
document.getElementById("objProf").innerHTML = "";  
document.getElementById("lang").innerHTML = "";  
document.getElementById("lang").innerHTML = "You learn no new languages";  
document.getElementById("objLang").innerHTML = "";  
document.getElementById("equipment").innerHTML = "";  
document.getElementById("equipment").innerHTML = "You receive: A small  
knife, a map of the city you grew up" +  
    "in, a pet mouse, a token to remember your parents by, a set  
of common clothes and 10 gold pieces";  
document.getElementById("objEquip").innerHTML = "";  
document.getElementById("feature").innerHTML = "";  
document.getElementById("feature").innerHTML = "City secrets";  
}
```

```

acolyte.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the acolyte specific background choices and alignment page, that sets the
choices to the coursework object. --&gt;
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;

&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
&lt;h2&gt;
    Background
&lt;/h2&gt;
&lt;%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}
if(flag){//if an alignment is chosen
</pre>

```

```

        session.setAttribute("player4", player); //set the latest version of the
object
        response.sendRedirect("backstory.jsp"); //redirect
    }
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into columns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality" id = "personality">
<option value = "I idolize a particular hero of my faith, and constantly refer to
that person's deeds and example.">
I idolize a particular hero of my faith, and constantly refer to that person's
deeds and example.</option>
<option value = "I can find common ground between the fiercest enemies,
empathizing with them and always working toward peace">
I can find common ground between the fiercest enemies, empathizing with them and
always working toward peace</option>
<option value = "I see omens in every event and action. The gods try to speak to
us, we just need to listen.">
I see omens in every event and action. The gods try to speak to us, we just need
to listen.</option>
<option value = "Nothing can shake my optimistic attitude">
Nothing can shake my optimistic attitude</option>
<option value = "I quote(or misquote) sacred texts and proverbs in almost every
situation">
I quote(or misquote) sacred texts and proverbs in almost every situation</option>
<option value = "I am tolerant(or intolerant) of other faithes and respect(or
condemn) the worship of other gods">
I am tolerant(or intolerant) of other faithes and respect(or condemn) the worship
of other gods</option>
<option value = "I've enjoyed fine food, drink, and high society among my temple's
elite. Rough Living grates on me">
I've enjoyed fine food, drink, and high society among my temple's elite. Rough
living grates on me</option>
<option value = "I've spent so long in the temple that I have little practical
experience dealing with people in the outside world.">
I've spent so long in the temple that I have little practical experience dealing
with people in the outside world.</option>
</select><br>
```

```
<b>Ideal</b><br>
<select name = "ideal">
<option value = "Tradition. The ancient traditions of worship and sacrifice must be preserved and upheld">
Tradition. The ancient traditions of worship and sacrifice must be preserved and upheld</option>
<option value = "Charity. I always try to help those in need, no matter what the personal cost.">
Charity. I always try to help those in need, no matter what the personal cost.</option>
<option value = "Change. We must help bring about the changes the gods are constantly working in the world.">
Change. We must help bring about the changes the gods are constantly working in the world.</option>
<option value = "Power. I hope to one day rise to the top of my faith's religious hierarchy">
Power. I hope to one day rise to the top of my faith's religious hierarchy</option>
<option value = "Faith. I trust that my deity will guide my actions. I have faith that if I work hard, things will go well.">
Faith. I trust that my deity will guide my actions. I have faith that if I work hard, things will go well.</option>
<option value = "Aspiration. I seek to prove myself worthy of my god's favor by matching my actions against his or her own teachings.">
Aspiration. I seek to prove myself worthy of my god's favor by matching my actions against his or her own teachings.</option>
</select><br>
```

```
<b>Bond</b><br>
<select name = "bond">
<option value = "I would die to recover an ancient relic of my faith that was lost long ago.">
I would die to recover an ancient relic of my faith that was lost long ago.</option>
<option value = "I will someday get revenge on the corrupt temple hierarchy who branded me a heretic">
I will someday get revenge on the corrupt temple hierarchy who branded me a heretic.</option>
<option value = "I owe my life to the priest who took me in when my parents died.">
I owe my life to the priest who took me in when my parents died.</option>
<option value = "Everything I do is for the common people">
Everything I do is for the common people</option>
<option value = "I will do anything to protect the temple where I served">
I will do anything to protect the temple where I served</option>
<option value = "I seek to preserve a sacred text that my enemies consider heretical and seek to destroy">
I seek to preserve a sacred text that my enemies consider heretical and seek to destroy.</option>
</select><br>
```

```
<b>Flaw</b><br>
<select name = "flaw">
<option value = "I judge others harshly, and myself even more severely">
I judge others harshly, and myself even more severely</option>
```

```
<option value = "I put too much trust in those who wield power within my temple's hierarchy">
I put too much trust in those who wield power within my temple's hierarchy</option>
<option value = "My piety sometimes leads me to blindly trust those that profess faith in my god.">
My piety sometimes leads me to blindly trust those that profess faith in my god.</option>
<option value = "I am inflexible in my thinking">
I am inflexible in my thinking</option>
<option value = "I am suspicous of strangers and expect the worst of them.">
I am suspicous of strangers and expect the worst of them.</option>
<option value = "Once I pick a goal, I become obsessed with it to the detriment of everything else in my life">
Once I pick a goal, I become obsessed with it to the detriment of everything else in my life</option>
</select><br><br>
<input type="submit" id = "fixedbutton">
</div>
</form>

</body>
</html>
```

```

charlatan.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the charlatan specific background choices and alignment page, that sets the
choices to the coursework object. --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;

&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
    &lt;h2&gt;
        Background
    &lt;/h2&gt;
&lt;%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}
</pre>

```

```

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into collumns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality">
<option value = "I fall in and out of love easily, and am always pursuing
someone.">
I fall in and out of love easily, and am always pursuing someone.</option>
<option value = "I have a joke for every occasion, especially occasions where
humour is inappropriate">
I have a joke for every occasion, especially occasions where humour is
inappropriate</option>
<option value = "Flattery is my preferred trick for getting what I want.">
Flattery is my preferred trick for getting what I want.</option>
<option value = "I'm a born gambler who can't resist taking a risk for a potential
payoff.">
I'm a born gambler who can't resist taking a risk for a potential payoff.</option>
<option value = "I lie about almost everything, even when there's no good reason
to.">
I lie about almost everything, even when there's no good reason to.</option>
<option value = "Sarcasm and insults are my weapons of choice.">
Sarcasm and insults are my weapons of choice.</option>
<option value = "I keep multiple holy symbols on me and invoke whatever deity
might come in useful at any given moment.">
I keep multiple holy symbols on me and invoke whatever deity might come in useful
at any given moment.</option>
<option value = "I pocket anything I see that might have some value.">
I pocket anything I see that might have some value.</option>
</select><br>

<b>Ideal</b><br>
<select name = "ideal">
<option value = "Independence. I am a free spirit - no one tells me what to do.">
Independence. I am a free spirit - no one tells me what to do.</option>

```

```

<option value = "Fairness. I never target people who can't afford to lose a few coins.">
Fairness. I never target people who can't afford to lose a few coins.</option>
<option value = "Charity. I distribute the money I acquire to the people who really need it.">
Charity. I distribute the money I acquire to the people who really need it.</option>
<option value = "Creativity. I never run the same con twice.">
Creativity. I never run the same con twice.</option>
<option value = "Friendship. Material goods come and go. Bonds of friendship last forever.">
Friendship. Material goods come and go. Bonds of friendship last forever.</option>
<option value = "Aspiration. I'm determined to make something of myself.">
Aspiration. I'm determined to make something of myself.</option>
</select><br>
```

```

<b>Bond</b><br>
<select name = "bond">
<option value = "I fleeced the wrong person and must work to ensure that this individual never crosses paths with me or those I care about.">
I fleeced the wrong person and must work to ensure that this individual never crosses paths with me or those I care about.</option>
<option value = "I owe everything to my mentor - a horrible person who's probably rotting in jail somewhere.">
I owe everything to my mentor - a horrible person who's probably rotting in jail somewhere.</option>
<option value = "Somewhere out there, I have a child who deosn't know me. I'm making the world better for him or her.">
Somewhere out there, I have a child who deosn't know me. I'm making the world better for him or her.</option>
<option value = "I come from a noble family, and one day I'll reclaim my Lands and title from those who stole them from me.">
I come from a noble family, and one day I'll reclaim my lands and title from those who stole them from me.</option>
<option value = "A ppowerful person killed someone I love. Some day soon, I'll have my revenge.">
A powerful person killed someone I love. Some day soon, I'll have my revenge.</option>
<option value = "I swindled and ruined a person who didn't deserve it. I seek to atone for my misdeeds but might never be able to forgive myself.">
I swindled and ruined a person who didn't deserve it. I seek to atone for my misdeeds but might never be able to forgive myself.</option>
</select><br>
```

```

<b>Flaw</b><br>
<select name = "flaw">
<option value = "I can't resist a pretty face.">
I can't resist a pretty face.</option>
<option value = "I'm always in debt. I spend my ill-gotten gains on decadent Luxuries faster than I bring them in...">
I'm always in debt. I spend my ill-gotten gains on decadent luxuries faster than I bring them in...</option>
<option value = "I'm convinced that no one could ever fool me the way I fool others">
I'm convinced that no one could ever fool me the way I fool others</option>
```

```
<option value = "I'm too greedy for my own good. I can't resist taking a risk if  
there's money involved.">  
I'm too greedy for my own good. I can't resist taking a risk if there's money  
involved.</option>  
<option value = "I can't resist swindling people who are more powerful than me.">  
I can't resist swindling people who are more powerful than me.</option>  
<option value = "I hate to admit it and will hate myself for it, but I'll run and  
preserve my own hide if the going gets tough.">  
I hate to admit it and will hate myself for it, but I'll run and preserve my own  
hide if the going gets tough.</option>  
</select><br><br>  
  
<input type="submit" id = "fixedbutton">  
</div>  
</form>  
  
</body>  
</html>
```

```

criminal.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the criminal specific background choices and alignment page, that sets the
choices to the coursework object. --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;

&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
    &lt;h2&gt;
        Background
    &lt;/h2&gt;
&lt;%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}
</pre>

```

```

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into collumns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality">
<option value = "I always have a plan for what to do when things go wrong.">
I always have a plan for what to do when things go wrong.</option>
<option value = "I am always calm, no matter what the situation. I never raise my
voice or let my emotions control me.">
I am always calm, no matter what the situation. I never raise my voice or let my
emotions control me.</option>
<option value = "The first thing I do in a new place is note the locations of
everything valuable - or where such things could be hidden.">
The first thing I do in a new place is note the locations of everything valuable -
or where such things could be hidden.</option>
<option value = "I would rather make a new friend than a new enemy.">
I would rather make a new friend than a new enemy.</option>
<option value = "I am incredibly slow to trust. Those who seem the fairest often
have the most to hide.">
I am incredibly slow to trust. Those who seem the fairest often have the most to
hide.</option>
<option value = "I don't pay attention to the risks in a situation. Never tell me
the odds.">
I don't pay attention to the risks in a situation. Never tell me the
odds.</option>
<option value = "The best way to get me to do something is to tell me I can't do
it.">
The best way to get me to do something is to tell me I can't do it.</option>
<option value = "I blow up at the slightest insult.">
I blow up at the slightest insult.</option>
</select><br>

<b>Ideal</b><br>
<select name = "ideal">

```

```

<option value = "Honor. I don't steal from others in the trade.">
Honor. I don't steal from others in the trade.</option>
<option value = "Freedom. Chains are meant to be broken, as are those who would
forge them.">
Freedom. Chains are meant to be broken, as are those who would forge
them.</option>
<option value = "Charity. I steal from the wealthy so that I can help people in
need.">
Charity. I steal from the wealthy so that I can help people in need.</option>
<option value = "Greed. I will do whatever it takes to become wealthy.">
Greed. I will do whatever it takes to become wealthy.</option>
<option value = "People. I'm loyal to my friends, not to any ideals, and everyone
else can take a trip down the Styx for all I care.">
People. I'm loyal to my friends, not to any ideals, and everyone else can take a
trip down the Styx for all I care.</option>
<option value = "Redemption. There's a spark of good in everyone.">
Redemption. There's a spark of good in everyone.</option>
</select><br>
```

```

<b>Bond</b><br>
<select name = "bond">
<option value = "I'm trying to pay off an old debt I owe to a generous
benefactor.">
I'm trying to pay off an old debt I owe to a generous benefactor.</option>
<option value = "My ill-gotten gains go to support my family">
My ill-gotten gains go to support my family</option>
<option value = "Something important was taken from me, and I aim to steal it
back.">
Something important was taken from me, and I aim to steal it back.</option>
<option value = "I will become the greatest thief that ever lived.">
I will become the greatest thief that ever lived.</option>
<option value = "I'm guilty of a terrible crime. I hope I can redeem myself for
it.">
I'm guilty of a terrible crime. I hope I can redeem myself for it.</option>
<option value = "Someone I loved died because of I mistake I made. That will never
happen again.">
Someone I loved died because of I mistake I made. That will never happen
again.</option>
</select><br>
```

```

<b>Flaw</b><br>
<select name = "flaw">
<option value = "When I see something valuable, I can't think about anything but
how to steal it.">
When I see something valuable, I can't think about anything but how to steal
it.</option>
<option value = "When faced with a choice between money and my friends, I usually
choose the money.">
When faced with a choice between money and my friends, I usually choose the
money.</option>
<option value = "If there's a plan, I'll forget it. If I don't forget it, I'll
ignore it.">
If there's a plan, I'll forget it. If I don't forget it, I'll ignore it.</option>
<option value = "I have a 'tell' that reveals when I'm lying.">
I have a 'tell' that reveals when I'm lying.</option>
<option value = "I turn tail and run when things look bad.">
```

```
I turn tail and run when things look bad.</option>
<option value = "An innocent person is in prison for a crime that I committed. I'm
okay with that.">
An innocent person is in prison for a crime that I committed. I'm okay with
that.</option>
</select><br><br>

<input type="submit" id = "fixedbutton">
</div>
</form>

</body>
</html>
```

```

entertainer.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the entertainer specific background choices and alignment page, that sets the
choices to the coursework object. --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;

&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
    &lt;h2&gt;
        Background
    &lt;/h2&gt;
&lt;%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}
</pre>

```

```

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into collumns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality">
<option value = "I know a story relevant to almost every situation.">
I know a story relevant to almost every situation.</option>
<option value = "Whenever I come to a new place, I collect local rumors and spread
gossip.">
Whenever I come to a new place, I collect local rumors and spread gossip.</option>
<option value = "I'm a hopeless romantic, always searching for that 'special
someone'.">
I'm a hopeless romantic, always searching for that 'special someone'.</option>
<option value = "Nobody stays angry at me or around me for long, since I can
diffuse any amount of tension.">
Nobody stays angry at me or around me for long, since I can diffuse any amount of
tension.</option>
<option value = "I love a good insult, even one directed at me.">
I love a good insult, even one directed at me.</option>
<option value = "I get bitter if I'm not the center of attention.">
I get bitter if I'm not the center of attention.</option>
<option value = "I'll settle for nothing less than perfection.">
I'll settle for nothing less than perfection.</option>
<option value = "I change my mood or mind as quickly as I change key in a song.">
I change my mood or mind as quickly as I change key in a song.</option>
</select><br>

<b>Ideal</b><br>
<select name = "ideal">
<option value = "Beauty. When I perform, I make the world better than it was.">
Beauty. When I perform, I make the world better than it was.</option>
<option value = "Tradition. The stories, legends, and songs of the past must never
be forgotten, for they teach us who we are.">

```

Tradition. The stories, legends, and songs of the past must never be forgotten, for they teach us who we are.</option>

<option value = "Creativity. The world is in need of new ideas and bold action."> Creativity. The world is in need of new ideas and bold action.</option>

<option value = "Greed. I'm only in it for the money and fame."> Greed. I'm only in it for the money and fame.</option>

<option value = "People. I like seeing the smiles on people's faces when I perform. That's all that matters."> People. I like seeing the smiles on people's faces when I perform. That's all that matters.</option>

<option value = "Honesty. Art should reflect the soul; it should come from within and reveal who we really are."> Honesty. Art should reflect the soul; it should come from within and reveal who we really are.</option>

</select><br>

<b>Bond</b><br>

<select name = "bond">

<option value = "My instrument is my most treasured possession, and it reminds me of someone I love."> My instrument is my most treasured possession, and it reminds me of someone I love.</option>

<option value = "Someone stole my precious instrument, and someday I'll get it back."> Someone stole my precious instrument, and someday I'll get it back.</option>

<option value = "I want to be famous, whatever it takes."> I want to be famous, whatever it takes.</option>

<option value = "I idolize a hero of the old tales and measure my deeds against that person's."> I idolize a hero of the old tales and measure my deeds against that person's.</option>

<option value = "I will do anything to prove myself superior to my hated rival."> I will do anything to prove myself superior to my hated rival.</option>

<option value = "I would do anything for the other members of my old troupe."> I would do anything for the other members of my old troupe.</option>

</select><br>

<b>Flaw</b><br>

<select name = "flaw">

<option value = "I'll do anything to win fame and renown."> I'll do anything to win fame and renown.</option>

<option value = "I'm a sucker for a pretty face."> I'm a sucker for a pretty face.</option>

<option value = "A scandal prevents me from ever going home again. That kind of trouble seems to follow me around."> A scandal prevents me from ever going home again. That kind of trouble seems to follow me around.</option>

<option value = "I once satirized a noble who still wants my head. It was a mistake that I will likely repeat."> I once satirized a noble who still wants my head. It was a mistake that I will likely repeat.</option>

<option value = "I have trouble keeping my true feelings hidden. My sharp tongue lands me in trouble."> I have trouble keeping my true feelings hidden. My sharp tongue lands me in trouble.</option>

<option value = "Despite my best efforts, I am unreliable to my friends.">

```
Despite my best efforts, I am unreliable to my friends.</option>
</select><br><br>

<input type="submit" id = "fixedbutton">
</div>
</form>

</body>
</html>
```

```

folkHero.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the folk hero specific background choices and alignment page, that sets the
choices to the coursework object. --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;

&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
    &lt;h2&gt;
        Background
    &lt;/h2&gt;
&lt;%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}
</pre>

```

```

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into collumns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality">
<option value = "I judge people by their actions, not their words.">
I judge people by their actions, not their words.</option>
<option value = "If someone is in trouble, I'm always ready to lend help.">
If someone is in trouble, I'm always ready to lend help.</option>
<option value = "When I set my mind to something, I follow through no matter what
gets in my way.">
When I set my mind to something, I follow through no matter what gets in my
way.</option>
<option value = "I have a strong sense of fair play and always try to find the
most equitable solution to arguments.">
I have a strong sense of fair play and always try to find the most equitable
solution to arguments.</option>
<option value = "I'm confident in my own abilities and do what I can to instill
confidence in others.">
I'm confident in my own abilities and do what I can to instill confidence in
others.</option>
<option value = "Thinking is for other people. I prefer action.">
Thinking is for other people. I prefer action.</option>
<option value = "I misuse long words in an attempt to sound smarter.">
I misuse long words in an attempt to sound smarter.</option>
<option value = "I get bored easily. When am I going to get on with my destiny?">
I get bored easily. When am I going to get on with my destiny?</option>
</select><br>

<b>Ideal</b><br>
<select name = "ideal">
<option value = "Respect. People deserve to be treated with dignity and respect.">
Respect. People deserve to be treated with dignity and respect.</option>

```

```
<option value = "Fairness. No one should get preferential treatment before the  
Law, and no one is above the Law.">  
Fairness. No one should get preferential treatment before the law, and no one is  
above the law.</option>  
<option value = "Freedom. Tyrants must not be allowed to oppress the people.">  
Freedom. Tyrants must not be allowed to oppress the people.</option>  
<option value = "Might. If I become strong, I can take what I want - what I  
deserve.">  
Might. If I become strong, I can take what I want - what I deserve.</option>  
<option value = "Sincerity. There's no good in pretending to be something I'm  
not.">  
Sincerity. There's no good in pretending to be something I'm not.</option>  
<option value = "Destiny. Nothing and no one can steer me away from my higher  
calling.">  
Destiny. Nothing and no one can steer me away from my higher calling.</option>  
</select><br>
```

```
<b>Bond</b><br>  
<select name = "bond">  
<option value = "I have a family, but I have no idea where they are. One day, I  
hope to see them again.">  
I have a family, but I have no idea where they are. One day, I hope to see them  
again.</option>  
<option value = "I worked the Land, I Love the Land, and I will protect the  
Land.">  
I worked the land, I love the land, and I will protect the land.</option>  
<option value = "A proud noble once gave me a horrible beating, and I will take my  
revenge on any bully I encounter.">  
A proud noble once gave me a horrible beating, and I will take my revenge on any  
bully I encounter.</option>  
<option value = "My tools are symbols of my past life, and I carry them so that I  
will never forget my roots.">  
My tools are symbols of my past life, and I carry them so that I will never forget  
my roots.</option>  
<option value = "I protect those who cannot protect themselves.">  
I protect those who cannot protect themselves.</option>  
<option value = "I wish my childhood sweetheart had come with me to pursue my  
destiny.">  
I wish my childhood sweetheart had come with me to pursue my destiny.</option>  
</select><br>
```

```
<b>Flaw</b><br>  
<select name = "flaw">  
<option value = "The tyrant who rules my Land will stop at nothing to see me  
killed.">  
The tyrant who rules my land will stop at nothing to see me killed.</option>  
<option value = "I'm convinced of the significance of my destiny, and blind to my  
shortcomings and the risk of failure.">  
I'm convinced of the significance of my destiny, and blind to my shortcomings and  
the risk of failure.</option>  
<option value = "The people who knew me when I was young know my shameful secret,  
so I can never go home again.">  
The people who knew me when I was young know my shameful secret, so I can never go  
home again.</option>  
<option value = "I have a weakness for the vices of the city, especially hard  
drink.">
```

```
I have a weakness for the vices of the city, especially hard drink.</option>
<option value = "Secretly, I believe that things would be better if I were a
tyrant lording over the land.">
Secretly, I believe that things would be better if I were a tyrant lording over
the land.</option>
<option value = "I have trouble trusting my allies.">
I have trouble trusting my allies.</option>
</select><br><br>

<input type="submit" id = "fixedbutton">
</div>
</form>

</body>
</html>
```

```

guildArtisan.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the guild artisan specific background choices and alignment page, that sets the
choices to the coursework object. --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;

&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
    &lt;h2&gt;
        Background
    &lt;/h2&gt;
&lt;%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}
</pre>

```

```

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into collumns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality">
<option value = "I believe that anything worth doing is worth doing right. I can't
help it - I'm a perfectionist.">
I believe that anything worth doing is worth doing right. I can't help it - I'm a
perfectionist.</option>
<option value = "I'm a snob who looks down on those who can't appreciate fine
art.">
I'm a snob who looks down on those who can't appreciate fine art.</option>
<option value = "I always want to know how things work and what makes people
tick.">
I always want to know how things work and what makes people tick.</option>
<option value = "I'm full of witty aphorisms and have a proverb for every
occasion.">
I'm full of witty aphorisms and have a proverb for every occasion.</option>
<option value = "I'm rude to people who lack my commitment to hard work and fair
play.">
I'm rude to people who lack my commitment to hard work and fair play.</option>
<option value = "I like to talk at length about my profession.">
I like to talk at length about my profession.</option>
<option value = "I don't part with my money easily and will haggle tireless to get
the best deal possible.">
I don't part with my money easily and will haggle tireless to get the best deal
possible.</option>
<option value = "I'm well known for my work, and I want to make sure everyone
appreciates it. I'm always taken aback when people haven't heard of me.">
I'm well known for my work, and I want to make sure everyone appreciates it. I'm
always taken aback when people haven't heard of me.</option>
</select><br>

<b>Ideal</b><br>

```

```
<select name = "ideal">
<option value = "Community. It is the duty of all civilised people to strengthen
the bonds of community and the security of civilisation.">
Community. It is the duty of all civilised people to strengthen the bonds of
community and the security of civilisation.</option>
<option value = "Generosity. My talents were given to me so that I could use them
to benefit the world.">
Generosity. My talents were given to me so that I could use them to benefit the
world.</option>
<option value = "Freedom. Everyone should be free to pursue his or her
livelihood.">
Freedom. Everyone should be free to pursue his or her livelihood.</option>
<option value = "Greed. I'm only in it for the money.">
Greed. I'm only in it for the money.</option>
<option value = "People. I'm committed to the people I care about, not to ideals.">
People. I'm committed to the people I care about, not to ideals.</option>
<option value = "Aspiration. I work hard to be the best there is at my craft.">
Aspiration. I work hard to be the best there is at my craft.</option>
</select><br>
```

```
<b>Bond</b><br>
<select name = "bond">
<option value = "The workshop where I Learned my trade is the most important place
in the world to me.">
The workshop where I learned my trade is the most important place in the world to
me.</option>
<option value = "I created a great work for someone, and then found them unworthy
to recieve it. I'm still looking for someone worthy.">
I created a great work for someone, and then found them unworthy to receive it.
I'm still looking for someone worthy.</option>
<option value = "I owe my guild a great debt for forging me into the person I am
today.">
I owe my guild a great debt for forging me into the person I am today.</option>
<option value = "I pursue wealth to secure someone's love.">
I pursue wealth to secure someone's love.</option>
<option value = "One day I will return to my guild and prove that I am the
greatest artisan of them all.">
One day I will return to my guild and prove that I am the greatest artisan of them
all.</option>
<option value = "I will get revenge on the evil forces that destroyed my place of
buisness and ruined my livelihood.">
I will get revenge on the evil forces that destroyed my place of buisness and
ruined my livelihood.</option>
</select><br>
```

```
<b>Flaw</b><br>
<select name = "flaw">
<option value = "I'll do anything to get my hands on something rare or
priceless.">
I'll do anything to get my hands on something rare or priceless.</option>
<option value = "I'm quick to assume that someone is trying to cheat me.">
I'm quick to assume that someone is trying to cheat me.</option>
<option value = "No one must ever learn that I once stole money from guild
coffers.">
No one must ever learn that I once stole money from guild coffers.</option>
<option value = "I'm never satisfied with what I have - I always want more.">
I'm never satisfied with what I have - I always want more.</option>
```

```
<option value = "I would kill to aquire a noble title.">  
I would kill to aquire a noble title.</option>  
<option value = "I'm horribly jealous of anyone who can outshine my handiwork.  
Everywhere I go, I'm surrounded by rivals.">  
I'm horribly jealous of anyone who can outshine my handiwork. Everywhere I go, I'm  
surrounded by rivals.</option>  
</select><br><br>  
  
<input type="submit" id = "fixedbutton">  
</div>  
</form>  
  
</body>  
</html>
```

```

hermit.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the hermit specific background choices and alignment page, that sets the
choices to the coursework object. --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;

&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
    &lt;h2&gt;
        Background
    &lt;/h2&gt;
&lt;%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}
</pre>

```

```

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into collumns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality">
<option value = "I've been isolated for so long that I rarely speak, preffering
gestures and the occasional grunt.">
I've been isolated for so long that I rarely speak, preferring gestures and the
occasional grunt.</option>
<option value = "I am utterly serene, even in the face of disaster.">
I am utterly serene, even in the face of disaster.</option>
<option value = "The Leader of my community had something wise to say on every
topic, I am eager to share that wisdom.">
The leader of my community had something wise to say on every topic, I am eager to
share that wisdom.</option>
<option value = "I feel tremendous empathy for all who suffer.">
I feel tremendous empathy for all who suffer.</option>
<option value = "I'm oblivious to ettiquette and social expectations.">
I'm oblivious to etiquette and social expectations.</option>
<option value = "I connect everything that happens to me to a grand, cosmic
plan.">
I connect everything that happens to me to a grand, cosmic plan.</option>
<option value = "I often get lost in my own thoughts and contemplation, becoming
oblivious to my surroundings.">
I often get lost in my own thoughts and contemplation, becoming oblivious to my
surroundings.</option>
<option value = "I am working on a grand philosophical theory and love sharing my
ideas.">
I am working on a grand philosophical theory and love sharing my ideas.</option>
</select><br>

<b>Ideal</b><br>
<select name = "ideal">

```

```
<option value = "Greater good. My gifts are meant to be shared with all, not used  
for my own benefit.">  
Greater good. My gifts are meant to be shared with all, not used for my own  
benefit.</option>  
<option value = "Logic. Emotions must not cloud our sense of what is right and  
true, or our logical thinking.">  
Logic. Emotions must not cloud our sense of what is right and true, or our logical  
thinking.</option>  
<option value = "Free Thinking. Inquiry and curiosity are the pillars of  
progress.">  
Free Thinking. Inquiry and curiosity are the pillars of progress.</option>  
<option value = "Power. Solitude and contemplation are paths towards mystical or  
magical power.">  
Power. Solitude and contemplation are paths towards mystical or magical  
power.</option>  
<option value = "Live and Let Live. Meddling in the affairs of others only causes  
trouble.">  
Live and Let Live. Meddling in the affairs of others only causes trouble.</option>  
<option value = "Self-Knowledge. If you know yourself, there's nothing left to  
know.">  
Self-Knowledge. If you know yourself, there's nothing left to know.</option>  
</select><br>
```

```
<b>Bond</b><br>  
<select name = "bond">  
<option value = "Nothing is more important than the other members of my hermitage,  
order, or association.">  
Nothing is more important than the other members of my hermitage, order, or  
association.</option>  
<option value = "I entered seclusion to hide from the ones who might still be  
hunting me. I must someday confront them.">  
I entered seclusion to hide from the ones who might still be hunting me. I must  
someday confront them.</option>  
<option value = "I'm still seeking the enlightenment I pursued in seclusion, and  
it still eludes me.">  
I'm still seeking the enlightenment I pursued in seclusion, and it still eludes  
me.</option>  
<option value = "I entered seclusion because I loved someone I could not have.">  
I entered seclusion because I loved someone I could not have.</option>  
<option value = "Should my discovery come to light, it could bring ruin to the  
world.">  
Should my discovery come to light, it could bring ruin to the world.</option>  
<option value = "My isolation gave me insight into a great evil that only I can  
destroy.">  
My isolation gave me insight into a great evil that only I can destroy.</option>  
</select><br>
```

```
<b>Flaw</b><br>  
<select name = "flaw">  
<option value = "Now that I've returned to the world, I enjoy its delights a  
little too much.">  
Now that I've returned to the world, I enjoy its delights a little too  
much.</option>  
<option value = "I harbor dark, bloodthirsty thoughts that my isolation and  
meditation failed to quell.">  
I harbor dark, bloodthirsty thoughts that my isolation and meditation failed to  
quell.</option>
```

```
<option value = "I am dogmatic in my thoughts and philosophy.">  
I am dogmatic in my thoughts and philosophy.</option>  
<option value = "I let my need to win arguments overshadow friendships and  
harmony.">  
I let my need to win arguments overshadow friendships and harmony.</option>  
<option value = "I'd risk too much to uncover a lost bit of knowledge.">  
I'd risk too much to uncover a lost bit of knowledge.</option>  
<option value = "I like keeping secrets and won't share them with anyone.">  
I like keeping secrets and won't share them with anyone.</option>  
</select><br><br>  
  
<input type="submit" id = "fixedbutton">  
</div>  
</form>  
  
</body>  
</html>
```

```

noble.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the noble specific background choices and alignment page, that sets the choices
to the coursework object. --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;

&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
    &lt;h2&gt;
        Background
    &lt;/h2&gt;
&lt;%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}
</pre>

```

```

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into collumns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality">
<option value = "My eloquent flattery makes everyone I talk to feel like the most
wonderful and important person in the world.">
    My eloquent flattery makes everyone I talk to feel like the most wonderful and
important person in the world.</option>
<option value = "The common folk love me for my kindness and generosity.">
    The common folk love me for my kindness and generosity.</option>
<option value = "No one could doubt me looking at my regal bearing that I am a cut
above the unwashed masses.">
    No one could doubt me looking at my regal bearing that I am a cut above the
unwashed masses.</option>
<option value = "I take great pains to always look my best and follow the latest
fashions.">
    I take great pains to always look my best and follow the latest fashions.</option>
<option value = "I don't like to get my hands dirty, and I won't be caught dead in
unsuitable accommodations.">
    I don't like to get my hands dirty, and I won't be caught dead in unsuitable
accommodations.</option>
<option value = "Despite my noble birth, I do not place myself above other folk.
we all have the same blood.">
    Despite my noble birth, I do not place myself above other folk. we all have the
same blood.</option>
<option value = "My favor, once lost, is lost forever.">
    My favor, once lost, is lost forever.</option>
<option value = "If you do me an injury, I will crush you, ruin your name, and
salt your fields.">
    If you do me an injury, I will crush you, ruin your name, and salt your
fields.</option>
</select><br>

```

```
<b>Ideal</b><br>
<select name = "ideal">
<option value = "Respect. respect is due to me because of my position, but all people regardless of station deserve to be treated with dignity.">
Respect. respect is due to me because of my position, but all people regardless of station deserve to be treated with dignity.</option>
<option value = "Responsibility. It is my duty to respect the authority of those above me, just as those below me must respect mine.">
Responsibility. It is my duty to respect the authority of those above me, just as those below me must respect mine.</option>
<option value = "Independence. I must prove that I can handle myself without the coddling of my family.">
Independence. I must prove that I can handle myself without the coddling of my family.</option>
<option value = "Power. If I can attain more power no one will tell me what to do.">
Power. If I can attain more power no one will tell me what to do.</option>
<option value = "Family. Blood runs thicker than water.">
Family. Blood runs thicker than water.</option>
<option value = "Noble Obligation. It is my duty to protect and care for the people beneath me.">
Noble Obligation. It is my duty to protect and care for the people beneath me.</option>
</select><br>
```

```
<b>Bond</b><br>
<select name = "bond">
<option value = "I will face any challenge to win the aproval of my family.">
I will face any challenge to win the aproval of my family.</option>
<option value = "My house's alliance with another noble family must be sustained at all costs.">
My house's alliance with another noble family must be sustained at all costs.</option>
<option value = "Nothing is more important than the other members of my family.">
Nothing is more important than the other members of my family.</option>
<option value = "I am in love with the heir of a family that my family despises.">
I am in love with the heir of a family that my family despises.</option>
<option value = "My loyalty to my sovereign is unwavering.">
My loyalty to my sovereign is unwavering.</option>
<option value = "The common folk must see me as a hero of the people.">
The common folk must see me as a hero of the people.</option>
</select><br>
```

```
<b>Flaw</b><br>
<select name = "flaw">
<option value = "I secretly believe that everyone is beneath me.">
I secretly believe that everyone is beneath me.</option>
<option value = "I hide a truly scandalous secret that could ruin my family forever.">
I hide a truly scandalous secret that could ruin my family forever.</option>
<option value = "I too often hear veiled insults and threats in every word addressed to me, and I'm quick to anger.">
I too often hear veiled insults and threats in every word addressed to me, and I'm quick to anger.</option>
<option value = "I have an insatiable desire for carnal pleasures.">
I have an insatiable desire for carnal pleasures.</option>
<option value = "In fact, the world does revolve around me.">
```

```
In fact, the world does revolve around me.</option>
<option value = "By my words and actions, I often bring shame to my family.">
By my words and actions, I often bring shame to my family.</option>
</select><br><br>

<input type="submit" id = "fixedbutton">
</div>
</form>

</body>
</html>
```

## outlander.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the outlander specific background choices and alignment page, that sets the
choices to the coursework object. --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;

&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
    &lt;h2&gt;
        Background
    &lt;/h2&gt;
&lt;%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}&gt;</pre>
```

```

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into collumns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality">
<option value = "I'm driven by a wonderlust that led me away from home.">
I'm driven by a wonderlust that led me away from home.</option>
<option value = "I watch over my friends as if they were a litter of newborn
pups.">
I watch over my friends as if they were a litter of newborn pups.</option>
<option value = "I once ran twenty-five miles without stopping to warn to my clan
of an approaching orc horde. I'd do it again if I had to.">
I once ran twenty-five miles without stopping to warn to my clan of an approaching
orc horde. I'd do it again if I had to.</option>
<option value = "I have a lesson for every situation, drawn from observing
nature.">
I have a lesson for every situation, drawn from observing nature.</option>
<option value = "I plaace no stock in wealthy or well-mannered folk. Money and
manners won't save you from a hungry owlbear">
I plaace no stock in wealthy or well-mannered folk. Money and manners won't save
you from a hungry owlbear</option>
<option value = "I'm always picking things up, absently fiddling with them, and
sometimes accidentally breaking them.">
I'm always picking things up, absently fiddling with them, and sometimes
accidentally breaking them.</option>
<option value = "I feel far more comfortable around animals than people.">
I feel far more comfortable around animals than people.</option>
<option value = "I was, in fact raised by wolves.">
I was, in fact raised by wolves.</option>
</select><br>

<b>Ideal</b><br>
<select name = "ideal">

```

```

<option value = "Change. Life is like the seasons, in constant change, and we must
change with it.">
Change. Life is like the seasons, in constant change, and we must change with
it.</option>
<option value = "Greater Good. It is each person's responsibility to make the most
happiness for the whole tribe.">
Greater Good. It is each person's responsibility to make the most happiness for
the whole tribe.</option>
<option value = "Honor. If I dishonour myself, I dishonour my whole clan.">
Honor. If I dishonour myself, I dishonour my whole clan.</option>
<option value = "Might. The strongest are meant to rule.">
Might. The strongest are meant to rule.</option>
<option value = "Nature. The natural world is more important than all the
constructs of civilisation.">
Nature. The natural world is more important than all the constructs of
civilisation.</option>
<option value = "Glory. I must earn glory in battle, for myself and my clan.">
Glory. I must earn glory in battle, for myself and my clan.</option>
</select><br>
```

<b>Bond</b><br>

```

<select name = "bond">
<option value = "My family, clan, or tribe is the most important thing in my life,
even when they are far from me.">
My family, clan, or tribe is the most important thing in my life, even when they
are far from me.</option>
<option value = "An injury to the unspoiled wilderness of my home is an injury to
me.">
An injury to the unspoiled wilderness of my home is an injury to me.</option>
<option value = "I will bring terrible wrath down on the evildoers who destroyed
my homeland.">
I will bring terrible wrath down on the evildoers who destroyed my
homeland.</option>
<option value = "I am the last of my tribe, and it is up to me to ensure their
names enter legend.">
I am the last of my tribe, and it is up to me to ensure their names enter
legend.</option>
<option value = "I suffer from awful visions of a coming disaster and will do
anything to prevent it.">
I suffer from awful visions of a coming disaster and will do anything to prevent
it.</option>
<option value = "It is my duty to provide children to sustain my tribe.">
It is my duty to provide children to sustain my tribe.</option>
</select><br>
```

<b>Flaw</b><br>

```

<select name = "flaw">
<option value = "I am too enamored of ale, wine and other intoxicants.">
I am too enamored of ale, wine and other intoxicants.</option>
<option value = "There's no room for caution in a life lived to the fullest.">
There's no room for caution in a life lived to the fullest.</option>
<option value = "I remember every insult I've received and nurse a silent
resentment toward anyone who's ever wronged me.">
I remember every insult I've received and nurse a silent resentment toward anyone
who's ever wronged me.</option>
```

```
<option value = "I am slow to trust members of other races, tribes, and societies.">  
I am slow to trust members of other races, tribes, and societies.</option>  
<option value = "Violence is my answer to almost any challenge.">  
Violence is my answer to almost any challenge.</option>  
<option value = "Don't expect me to save those who can't save themselves. It is nature's way that the strong thrive and the weak perish.">  
Don't expect me to save those who can't save themselves. It is nature's way that the strong thrive and the weak perish.</option>  
</select><br><br>  
  
<input type="submit" id = "fixedbutton">  
</div>  
</form>  
  
</body>  
</html>
```

```

sage.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the sage specific background choices and alignment page, that sets the choices
to the coursework object. -->

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>

</head>
<body>
<h1>Bram's character creator</h1>
    <h2>
        Background
    </h2>
<%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}

```

```

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into collumns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevent personality traits, ideals, bonds and flaws as dropdown box options -->

<b>Personality trait</b><br>
<select name = "personality">
<option value = "I use polysyllabic words that convey the impression of great
erudition.">
I use polysyllabic words that convey the impression of great erudition.</option>
<option value = "I've read every book in the world's greatest libraries - or like
to boast that I have">
I've read every book in the world's greatest libraries - or like to boast that I
have</option>
<option value = "I'm used to helping out those who aren't as smart as I am, and I
patiently explain anything and everything to others.">
I'm used to helping out those who aren't as smart as I am, and I patiently explain
anything and everything to others.</option>
<option value = "There's nothing I like more than a good mystery.">
There's nothing I like more than a good mystery.</option>
<option value = "I'm willing to listen to every side of an argument before I make
my own judgement.">
I'm willing to listen to every side of an argument before I make my own
judgement.</option>
<option value = "I...speak...slowly...when talking...to idiots...
which...almost...everyone...is...compared...to me.">
I...speak...slowly...when talking...to idiots...
which...almost...everyone...is...compared...to me.</option>
<option value = "I am horribly, horribly awkward in social situations.">
I am horribly, horribly awkward in social situations.</option>
<option value = "I'm convinced that people are always trying to steal my
secrets.">
I'm convinced that people are always trying to steal my secrets.</option>
</select><br>

```

```
<b>Ideal</b><br>
<select name = "ideal">
<option value = "Knowledge. The path to power and self-improvement is through knowledge.">
Knowledge. The path to power and self-improvement is through knowledge.</option>
<option value = "Beauty. What is beautiful points us beyond itself towards what is true.">
Beauty. What is beautiful points us beyond itself towards what is true.</option>
<option value = "Logic. Emotions must not cloud our logical thinking.">
Logic. Emotions must not cloud our logical thinking.</option>
<option value = "No Limits. Nothing should fetter the infinite possibility inherent in all existence.">
No Limits. Nothing should fetter the infinite possibility inherent in all existence.</option>
<option value = "Power. Knowledge is the path to power and domination.">
Power. Knowledge is the path to power and domination.</option>
<option value = "Self-Improvement. The goal of a life of study is the betterment of oneself.">
Self-Improvement. The goal of a life of study is the betterment of oneself.</option>
</select><br>
```

```
<b>Bond</b><br>
<select name = "bond">
<option value = "It is my duty to protect my students.">
It is my duty to protect my students.</option>
<option value = "I have an ancient text that holds terrible secrets that must not fall into the wrong hands.">
I have an ancient text that holds terrible secrets that must not fall into the wrong hands.</option>
<option value = "I work to preserve a library, university, scriptorium or monastary.">
I work to preserve a library, university, scriptorium or monastary.</option>
<option value = "My life's work is a series of tomes related to a specific field of lore.">
My life's work is a series of tomes related to a specific field of lore.</option>
<option value = "I've been searching my whole life for the answer to a certain question.">
I've been searching my whole life for the answer to a certain question.</option>
<option value = "I sold my soul for knowledge. I hope to do great deeds and win it back.">
I sold my soul for knowledge. I hope to do great deeds and win it back.</option>
</select><br>
```

```
<b>Flaw</b><br>
<select name = "flaw">
<option value = "I am easily distracted by the promise of information.">
I am easily distracted by the promise of information.</option>
<option value = "Most people scream and run when they see a demon. I stop and take notes on its anatomy.">
Most people scream and run when they see a demon. I stop and take notes on its anatomy.</option>
<option value = "Unlocking an ancient mystery is worth the price of a civilisation.">
Unlocking an ancient mystery is worth the price of a civilisation.</option>
<option value = "I overlook obvious solutions in favor of complicated ones.">
```

```
I overlook obvious solutions in favor of complicated ones.</option>
<option value = "I speak without really thinking through my words, invariably
insulting others.">
I speak without really thinking through my words, invariably insulting
others.</option>
<option value = "I can't keep a secret to save my life, or anyone else's.">
I can't keep a secret to save my life, or anyone else's.</option>
</select><br><br>

<input type="submit" id = "fixedbutton">
</div>
</form>

</body>
</html>
```

## sailor.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the sailor specific background choices and alignment page, that sets the
choices to the coursework object. -->

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>

</head>
<body>
<h1>Bram's character creator</h1>
<h2>
    Background
</h2>
<%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
<hr>
```

```

}

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into columns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevent personality traits, ideals, bonds and flaws as dropdown box options -->

    <b>Personality trait</b><br> <select name="personality">

        <option value="My friends know they can rely on me, no
matter what.">
            My friends know they can rely on me, no matter
what.</option>

        <option
            value="I work hard so that I can play hard when
the work is done.">
            I work hard so that I can play hard when the work
is done.</option>

        <option
            value="I enjoy sailing into new ports and making
new friends over a flagon of ale.">
            I enjoy sailing into new ports and making new
friends over a flagon
            of ale.</option>

        <option
            value="I stretch the truth for the sake of a
good story.">
            I stretch the truth for the sake of a good
story.</option>

        <option
            value="To me, a tavern brawl is a nice way to get
to know a new city.">

```

```

To me, a tavern brawl is a nice way to get to
know a new city.</option>

<option value="I never pass up a friendly wager.">I
never
    pass up a friendly wager.</option>

<option value="My language is as foul as an otyugh
nest.">
    My language is as foul as an otyugh
nest.</option>

<option
    value="I like a job well done, especially if I
can convince someone else to do it.">
    I like a job well done, especially if I can
convince someone else
        to do it.</option>

</select><br> <b>Ideal</b><br> <select name="ideal">

<option
    value="Respect. The thing that keeps a ship
together is mutual respect between captain and crew.">
    Respect. The thing that keeps a ship together is
mutual respect
        between captain and crew</option>

<option
    value="Fairness. We all do the work, so we all
share in the rewards.">
    Fairness. We all do the work, so we all share in
the rewards.</option>

<option
    value="Freedom. The sea is freedom - the freedom
to go anywhere and do anything.">
    Freedom. The sea is freedom - the freedom to go
anywhere and do
        anything.</option>

<option
    value="Mastery. I'm a predator, and the other
ships on the sea are my prey.">
    Mastery. I'm a predator, and the other ships on
the sea are my
        prey.</option>

<option
    value="People. I'm committed to my crewmates, not
to ideals.">
    People. I'm committed to my crewmates, not to
ideals.</option>

<option
    value="Aspiration. Someday I'll own my own ship
and chart my own destiny.">

```

Aspiration. Someday I'll own my own ship and chart my own destiny.</option>

```

</select><br> <b>Bond</b><br> <select name="bond">

<option
    value="I'm loyal to my captain first, everything else second.">
        I'm loyal to my captain first, everything else second.</option>

<option
    value="The ship is most important - crewmaes and captains come and go.">
        The ship is most important - crewmaes and captains come and go.</option>

<option value="I'll always remember my first ship.">I'll
        always remember my first ship.</option>

<option
    value="In a harbor town, I have a paramour whose eyes nearly stole me from the sea.">
        In a harbor town, I have a paramour whose eyes nearly stole me from
            the sea.</option>

<option
    value="I was cheated out of my fair share of the profits, and I want to get my due.">
        I was cheated out of my fair share of the
            profits, and I want to
                get my due.</option>

<option
    value="Ruthless pirates murdered my captain and crewmates, plundered our ship, and left me to die. Vengeance will be mine.">
        Ruthless pirates murdered my captain and
            crewmates, plundered our
                ship, and left me to die. Vengeance will be
                    mine.</option>

</select><br> <b>Flaw</b><br> <select name="flaw">

<option value="I follow orders, even if I think they're wrong.">
        I follow orders, even if I think they're
            wrong.</option>

<option value="I'll say anything to avoid having to do extra work.">
        I'll say anything to avoid having to do extra
            work.</option>

<option
    value="Once someone questions my courage, I never back down no matter how dangerous the situation.">

```

```
down no matter how  
Once someone questions my courage, I never back  
dangerous the situation.</option>  
once I start drinking, it's hard for me to  
stop.</option>  
  
value="I can't help but pocket loose coins and  
other trinkets I come across.">  
I can't help but pocket loose coins and other  
trinkets I come  
across.</option>  
  
My pride will probably lead to my  
destruction.</option>  
  
</select><br> <br> <input type="submit" id="fixedbutton">  
</div>  
</form>  
</body>  
</html>
```

```

soldier.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the soldier specific background choices and alignment page, that sets the
choices to the coursework object. --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;

&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
    &lt;h2&gt;
        Background
    &lt;/h2&gt;
&lt;%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen
}
</pre>

```

```

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into collumns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality">
<option value = "I'm always polite and respectful.">
I'm always polite and respectful.</option>
<option value = "I'm haunted by memories of war.I can't get the images of violence
out of my mind.">
I'm haunted by memories of war.I can't get the images of violence out of my
mind.</option>
<option value = "I've lost too many friends, and I'm slow to make new ones.">
I've lost too many friends, and I'm slow to make new ones.</option>
<option value = "I'm full of inspiring and cautionary tales from my military
experience relevant to almost every">
I'm full of inspiring and cautionary tales from my military experience relevant to
almost every</option>
<option value = "I can stare down a hell hound without flinching.">
I can stare down a hell hound without flinching.</option>
<option value = "I enjoy being strong and like breaking things.">
I enjoy being strong and like breaking things.</option>
<option value = "I have a crude sense of humor.">
I have a crude sense of humor.</option>
<option value = "I face problems head-on. A simple, direct solution is the best
path to success.">
I face problems head-on. A simple, direct solution is the best path to
success.</option>
</select><br>

<b>Ideal</b><br>
<select name = "ideal">
<option value = "Greater Good. Our Lot is to lay down our lives in defence of
others.">
Greater Good. Our lot is to lay down our lives in defence of others.</option>

```

```

<option value = "Responsibility. I do what I must and obey just authority.">
Responsibility. I do what I must and obey just authority.</option>
<option value = "Independence. When people follow orders blindly, they embrace a
kind of tyranny.">
Independence. When people follow orders blindly, they embrace a kind of
tyranny.</option>
<option value = "Might. In life as in war, the stronger force wins.">
Might. In life as in war, the stronger force wins.</option>
<option value = "Live and Let Live. Ideals aren't worth killing over or going to
war for.">
Live and Let Live. Ideals aren't worth killing over or going to war for.</option>
<option value = "My city, nation, or people are all that matter.">
My city, nation, or people are all that matter.</option>
</select><br>
```

<b>Bond</b><br>

```

<select name = "bond">
<option value = "I would still lay down my life for the people I served with.">
I would still lay down my life for the people I served with.</option>
<option value = "Someone saved my life on the battlefield. To this day, I will
never leave a friend behind.">
Someone saved my life on the battlefield. To this day, I will never leave a friend
behind.</option>
<option value = "My honor is my life.">
My honor is my life.</option>
<option value = "I'll never forget the crushing defeat my company suffered or the
enemies who dealt it.">
I'll never forget the crushing defeat my company suffered or the enemies who dealt
it.</option>
<option value = "Those who fight beside me are those worth dying for.">
Those who fight beside me are those worth dying for.</option>
<option value = "I fight for those who cannot fight for themselves.">
I fight for those who cannot fight for themselves.</option>
</select><br>
```

<b>Flaw</b><br>

```

<select name = "flaw">
<option value = "The monstrous enemy we face in battle still leaves me quivering
with fear.">
The monstrous enemy we face in battle still leaves me quivering with
fear.</option>
<option value = "I have little respect for anyone who is not a proven warrior.">
I have little respect for anyone who is not a proven warrior.</option>
<option value = "I made a terrible mistake in battle that cost many lives - and I
would do anything to keep that mistake secret">
I made a terrible mistake in battle that cost many lives - and I would do anything
to keep that mistake secret</option>
<option value = "My hatred of my enemies is blind and unreasoning.">
My hatred of my enemies is blind and unreasoning.</option>
<option value = "I obey the law, even if the law causes misery.">
I obey the law, even if the law causes misery.</option>
<option value = "I'd rather eat my armor than admit when I'm wrong.">
I'd rather eat my armor than admit when I'm wrong.</option>
</select><br><br>
```

<input type="submit" id = "fixedbutton">

```
</div>
</form>

</body>
</html>
```

## urchin.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the urchin specific background choices and alignment page, that sets the
choices to the coursework object. -->

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>

</head>
<body>
<h1>Bram's character creator</h1>
<h2>
    Background
</h2>
<%
session.getAttribute("player3");//import latest version of object
coursework player = (coursework)session.getAttribute("player3");//set object to a
useable variable
int selectedItem = 0;//set to a value the user picks for alignment
boolean flag = false;
if (request.getParameter("align") != null) {
    selectedItem = Integer.parseInt(request.getParameter("align"));//gets the
users chosen
    player.setAlignment(selectedItem);//set the players alignment to the
appropriate value
    flag = true;//set continue flag
}
if(request.getParameter("personality") != null){
    player.personality = request.getParameter("personality");//set the players
personality to the value chosen
}
if(request.getParameter("ideal") != null){
    player.ideal = request.getParameter("ideal");//set the players ideal to the
value chosen
}
if(request.getParameter("bond") != null){
    player.bond = request.getParameter("bond");//set the players bond to the
value chosen
}
if(request.getParameter("flaw") != null){
    player.flaw = request.getParameter("flaw");//set the flaw personality to
the value chosen

```

```

}

if(flag){//if an alignment is chosen
    session.setAttribute("player4", player);//set the latest version of the
object
    response.sendRedirect("backstory.jsp");//redirect
}
%>
<form>
<div id = "align"><!-- radio buttons for choosing the players alignment -->
    <input type="radio" name="align" id="Chaotic good" value="0">Chaotic
good<!-- seperated into columns of good, neutral and chaotic -->
    <input type="radio" name="align" id="Good" value="1">Good
    <input type="radio" name="align" id="Lawful good" value="2">Lawful
good<br><!-- seperated into rows of good, neutral and evil -->
    <input type="radio" name="align" id="Chaotic neutral"
value="3">Chaotic neutral
    <input type="radio" name="align" id="Neutral" value="4">Neutral
    <input type="radio" name="align" id="Lawful neutral"
value="5">Lawful neutral <br>
    <input type="radio" name="align" id="Chaotic evil" value="6">Chaotic
evil
    <input type="radio" name="align" id="Evil" value="7">Evil
    <input type="radio" name="align" id="Lawful evil" value="8">Lawful
evil<br>

<h2><%= player.background %></h2><!-- output the players chosen background and all
relevant personality traits, ideals, bonds and flaws as dropdown box options -->
<b>Personality trait</b><br>
<select name = "personality">
<option value = "I hide scraps of food and trinkets away in my pockets.">
I hide scraps of food and trinkets away in my pockets.</option>
<option value = "I ask lots of questions.">
I ask lots of questions.</option>
<option value = "I like to squeeze into small places where no one else can get to
me.">
I like to squeeze into small places where no one else can get to me.</option>
<option value = "I sleep with my back to a wall or tree, with everything I own
wrapped in a bundle in my arms.">
I sleep with my back to a wall or tree, with everything I own wrapped in a bundle
in my arms.</option>
<option value = "I eat like a pig and have bad manners.">
I eat like a pig and have bad manners.</option>
<option value = "I think anyone who's mice to me is hiding evil intent.">
I think anyone who's mice to me is hiding evil intent.</option>
<option value = "I don't like to bathe.">
I don't like to bathe.</option>
<option value = "I bluntly say what other people are hinting at or hiding.">
I bluntly say what other people are hinting at or hiding.</option>
</select><br>

<b>Ideal</b><br>
<select name = "ideal">
<option value = "Respect. All people, rich or poor, deserve respect.">
Respect. All people, rich or poor, deserve respect.</option>
<option value = "Community. We have to take care of each other, because no one
else is going to do it.">

```

Community. We have to take care of each other, because no one else is going to do it.</option>

<option value = "Change. The low are lifted up, and the high and mighty are brought down. Change is the nature of things.">

Change. The low are lifted up, and the high and mighty are brought down. Change is the nature of things.</option>

<option value = "Retribution. The rich need to be shown what life and death are like in the gutters.">

Retribution. The rich need to be shown what life and death are like in the gutters.</option>

<option value = "People. I help the people who help me - that's what keeps us alive.">

People. I help the people who help me - that's what keeps us alive.</option>

<option value = "Aspiration. I'm going to prove that I'm worthy of a better life.">

Aspiration. I'm going to prove that I'm worthy of a better life.</option>

</select><br>

<b>Bond</b><br>

<select name = "bond">

<option value = "my town or city is my home, and I'll fight to defend it.">

my town or city is my home, and I'll fight to defend it.</option>

<option value = "I sponsor my orphanage to keep others from enduring what I was forced to endure.">

I sponsor my orphanage to keep others from enduring what I was forced to endure.</option>

<option value = "I owe my survival to another urchin who taught me to live on the streets.">

I owe my survival to another urchin who taught me to live on the streets.</option>

<option value = "I owe a debt I can never repay to the person who took pity on me.">

I owe a debt I can never repay to the person who took pity on me.</option>

<option value = "I escaped my life of poverty by robbing an important person, and I'm wanted for it.">

I escaped my life of poverty by robbing an important person, and I'm wanted for it.</option>

<option value = "No one else should have to endure the hardships I've been thorough.">

No one else should have to endure the hardships I've been thorough.</option>

</select><br>

<b>Flaw</b><br>

<select name = "flaw">

<option value = "If I'm outnumbered, I will run away from a fight.">

If I'm outnumbered, I will run away from a fight.</option>

<option value = "Gold seems like a lot of money to me, and I'll do just about anything for more of it.">

Gold seems like a lot of money to me, and I'll do just about anything for more of it.</option>

<option value = "I will never fully trust anyone other than myself.">

I will never fully trust anyone other than myself.</option>

<option value = "I'd rather kill someone in their sleep than fight fair.">

I'd rather kill someone in their sleep than fight fair.</option>

<option value = "It's not stealing if I need it more than someone else.">

It's not stealing if I need it more than someone else.</option>

<option value = "People who can't take care of themselves get what they deserve.">

```
People who can't take care of themselves get what they deserve.</option>
</select><br><br>

<input type="submit" id = "fixedbutton">
</div>
</form>
</body>
</html>
```

## backstory.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!-- a simple page that allows the backstory string to be added to --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
&lt;h2&gt;
    Backstory
&lt;/h2&gt;
&lt;%
session.getAttribute("player4");//get latest object
coursework player = (coursework)session.getAttribute("player4");//make useable
if (request.getParameter("backstory") != null) {
    if (request.getParameter("backstory") != ""){//check a backstory was
entered
        if(request.getParameter("backstory").length() &lt;490){//check the
backstory wasn't longer than 490 characters due to google datastore constraints
            try{
                player.backstory =
request.getParameter("backstory");//set player backstory to what was entered
                session.setAttribute("player5", player);
                response.sendRedirect("abilChoice.jsp");//redirect to
next page
            }catch(Exception e){
            }
        }else{
            out.println("backstory limmited to 490 characters");//output the
user entered too many characters
        }
    }
}
%&gt;
&lt;form&gt;
&lt;div id = "align"&gt;
&lt;textarea name="backstory" style="width: 300px; height:
450px;"&gt;&lt;/textarea&gt;&lt;br&gt;&lt;br&gt;&lt;!-- create text area for bakcstory input --&gt;
&lt;input type="submit" id = "fixedbutton"&gt;&lt;!-- submit button --&gt;
&lt;/div&gt;
&lt;/form&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```



## Ability score selection

### abilChoice.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!--the page for character ability score selection --&gt;
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;
&lt;%
    session.getAttribute("player4");
    coursework player = (coursework)session.getAttribute("player4");
%&gt;
&lt;h1&gt;Bram's character creator&lt;/h1&gt;
&lt;h2&gt;
    Ability Scores
&lt;/h2&gt;
&lt;div id = "align"&gt;&lt;!-- centralise all text --&gt;
&lt;center&gt;&lt;!-- centralise the buttons --&gt;
    &lt;a href="randAbil.jsp"&gt;&lt;button type="button"&gt;Random?&lt;/button&gt;&lt;/a&gt;&lt;!-- link
to the random ability score page --&gt;
&lt;/center&gt;
    Use numbers as though you had rolled dice and used those values&lt;br&gt;&lt;br&gt;
&lt;center&gt;
    &lt;a href ="preset.jsp"&gt;&lt;button type="button"&gt;Preset?&lt;/button&gt;&lt;/a&gt;&lt;!-- link
to the preset ability score page --&gt;
&lt;/center&gt;
    use the standard number&lt;br&gt;&lt;br&gt;
&lt;center&gt;
    &lt;a href ="pointBought.jsp"&gt;&lt;button type="button"&gt;Point
bought?&lt;/button&gt;&lt;/a&gt;&lt;!-- link to the point bought ability score page --&gt;
&lt;/center&gt;
    use custom numbers by spending points&lt;br&gt;&lt;br&gt;
&lt;/div&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>
```

```

randAbil.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the page for determining random ability scores -->

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">

<script src="validRadio.js"></script><!-- imports a function to make the radio
buttons work in two dimensions -->
<%
    session.getAttribute("player5");//import latest object
    coursework player = (coursework)session.getAttribute("player5");
    int point1 = processing.rand4D6();//generate the players 6 dice rolls for
ability scores
    int point2 = processing.rand4D6();
    int point3 = processing.rand4D6();
    int point4 = processing.rand4D6();
    int point5 = processing.rand4D6();
    int point6 = processing.rand4D6();
    int st = 0;//initialise the values that will be appended to the object
    int de = 0;
    int co = 0;
    int in = 0;
    int wi = 0;
    int ch = 0;
int check;//counter
boolean[] flag = {false, false, false, false, false};//array of flags to
check all 6 choices have been made
if (request.getParameter("str") != null) {
    st = Integer.parseInt(request.getParameter("str"));//set the str to the
value selected in the radio buttons for str
    flag[0] = true;//set first flag to true
}
if (request.getParameter("dex") != null) {
    de = Integer.parseInt(request.getParameter("dex"));//set the dex to the
chosen value
    flag[1] = true;//second flag true
}
if (request.getParameter("con") != null) {
    co = Integer.parseInt(request.getParameter("con"));//set the constitution
to the chosen value
    flag[2] = true;//third flag true
}
if (request.getParameter("int") != null) {
    in = Integer.parseInt(request.getParameter("int"));//set the intelligence
to the chosen value
    flag[3] = true;//fourth flag true
}
if (request.getParameter("wis") != null) {
    wi = Integer.parseInt(request.getParameter("wis"));//set the wisdom to the
chosen value
    flag[4] = true;//fifth flag true
}
if (request.getParameter("cha") != null) {

```

```

        ch = Integer.parseInt(request.getParameter("cha")); //set the charisma to
the chosen value
        flag[5] = true; //sixth flag true
    }

check = 0;
for(int x=0; x<6; x++){
    if(flag[x] == true){
        check++; //if all the flags are true this value will be 6
    }
    if(check == 6){ //if all the flags are true
        player.seta1(st); //add the chosen strength to the players current
strength
        player.seta2(de); //add the chosen dexterity
        player.seta3(co); //add the chosen constitution
        player.seta4(in); //add the chosen intelligence
        player.seta5(wi); //add the chosen wisdom
        player.seta6(ch); //add the chosen charisma
        session.setAttribute("player6", player); //set latest object
        response.sendRedirect("class.jsp"); //redirect
    }
}
%>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Random
</h2>
<form>
    <div id="control">
        <div id="leftcolumn1"><!-- left column -->
            <b>ability scores:</b>
            <table><!-- table for formatting -->
                <tr>
                    <td>Strength:</td>
                    <td><input type="radio" id = "one1" name = "str"
value="<%=point1 %>" onclick = "oneWipe(1);"><%=point1 %> </td> <!-- put each
random value as an option for each ability -->
                    <td><input type="radio" id = "two1" name = "str"
value="<%=point2 %>" onclick = "twoWipe(1);"><%=point2 %> </td> <!-- the function
makes each row act like a radio button as well as the columns -->
                    <td><input type="radio" id = "three1" name = "str"
value="<%=point3 %>" onclick = "threeWipe(1);"><%=point3 %> </td> <!-- the
parameter of the function tells the js which row is checked -->
                    <td><input type="radio" id = "four1" name = "str"
value="<%=point4 %>" onclick = "fourWipe(1);"><%=point4 %> </td>
                    <td><input type="radio" id = "five1" name = "str"
value="<%=point5 %>" onclick = "fiveWipe(1);"><%=point5 %> </td>

```

```

        <td><input type="radio" id ="six1" name = "str"
value="<%>point6 %>" onclick = "sixWipe(1);"><%=point6 %></td>
    </tr>
    <tr>
        <td>Dexterity: </td>
        <td><input type="radio" id ="one2" name = "dex"
value="<%>point1 %>" onclick = "oneWipe(2);"><%=point1 %></td>
            <td><input type="radio" id ="two2" name = "dex"
value="<%>point2 %>" onclick = "twoWipe(2);"><%=point2 %></td>
                <td><input type="radio" id ="three2" name = "dex"
value="<%>point3 %>" onclick = "threeWipe(2);"><%=point3 %></td>
                    <td><input type="radio" id ="four2" name = "dex"
value="<%>point4 %>" onclick = "fourWipe(2);"><%=point4 %></td>
                        <td><input type="radio" id ="five2" name = "dex"
value="<%>point5 %>" onclick = "fiveWipe(2);"><%=point5 %></td>
                            <td><input type="radio" id ="six2" name = "dex"
value="<%>point6 %>" onclick = "sixWipe(2);"><%=point6 %></td>
                                </tr>
                                <tr>
                                    <td>Constitution:</td>
                                    <td><input type="radio" id ="one3" name = "con"
value="<%>point1 %>" onclick = "oneWipe(3);"><%=point1 %></td>
                                        <td><input type="radio" id ="two3" name = "con"
value="<%>point2 %>" onclick = "twoWipe(3);"><%=point2 %></td>
                                            <td><input type="radio" id ="three3" name = "con"
value="<%>point3 %>" onclick = "threeWipe(3);"><%=point3 %></td>
                                                <td><input type="radio" id ="four3" name = "con"
value="<%>point4 %>" onclick = "fourWipe(3);"><%=point4 %></td>
                                                    <td><input type="radio" id ="five3" name = "con"
value="<%>point5 %>" onclick = "fiveWipe(3);"><%=point5 %></td>
                                                        <td><input type="radio" id ="six3" name =
"con" value="<%>point6 %>" onclick = "sixWipe(3);"><%=point6 %></td>
                                </tr>
                                <tr>
                                    <td>Intelligence:</td>
                                    <td><input type="radio" id ="one4" name = "int"
value="<%>point1 %>" onclick = "oneWipe(4);"><%=point1 %></td>
                                        <td><input type="radio" id ="two4" name = "int"
value="<%>point2 %>" onclick = "twoWipe(4);"><%=point2 %></td>
                                            <td><input type="radio" id ="three4" name = "int"
value="<%>point3 %>" onclick = "threeWipe(4);"><%=point3 %></td>
                                                <td><input type="radio" id ="four4" name = "int"
value="<%>point4 %>" onclick = "fourWipe(4);"><%=point4 %></td>
                                                    <td><input type="radio" id ="five4" name = "int"
value="<%>point5 %>" onclick = "fiveWipe(4);"><%=point5 %></td>
                                                        <td><input type="radio" id ="six4" name = "int"
value="<%>point6 %>" onclick = "sixWipe(4);"><%=point6 %></td>
                                </tr>
                                <tr>
                                    <td>Wisdom:</td>
                                    <td><input type="radio" id ="one5" name = "wis"
value="<%>point1 %>" onclick = "oneWipe(5);"><%=point1 %></td>
                                        <td><input type="radio" id ="two5" name = "wis"
value="<%>point2 %>" onclick = "twoWipe(5);"><%=point2 %></td>
                                            <td><input type="radio" id ="three5" name = "wis"
value="<%>point3 %>" onclick = "threeWipe(5);"><%=point3 %></td>
                                                <td><input type="radio" id ="four5" name = "wis"
value="<%>point4 %>" onclick = "fourWipe(5);"><%=point4 %></td>

```

```

        <td><input type="radio" id ="five5" name = "wis"
value="<%>point5 %>" onclick = "fiveWipe(5);"><%=point5 %></td>
        <td><input type="radio" id ="six5" name = "wis"
value="<%>point6 %>" onclick = "sixWipe(5);"><%=point6 %></td>
    </tr>
    <tr>
        <td>Charisma:</td>
        <td><input type="radio" id ="one6" name = "cha"
value="<%>point1 %>" onclick = "oneWipe(6);"><%=point1 %></td>
        <td><input type="radio" id ="two6" name = "cha"
value="<%>point2 %>" onclick = "twoWipe(6);"><%=point2 %></td>
        <td><input type="radio" id ="three6" name = "cha"
value="<%>point3 %>" onclick = "threeWipe(6);"><%=point3 %></td>
        <td><input type="radio" id ="four6" name = "cha"
value="<%>point4 %>" onclick = "fourWipe(6);"><%=point4 %></td>
        <td><input type="radio" id ="five6" name = "cha"
value="<%>point5 %>" onclick = "fiveWipe(6);"><%=point5 %></td>
        <td><input type="radio" id ="six6" name = "cha"
value="<%>point6 %>" onclick = "sixWipe(6);"><%=point6 %></td>
    </tr>
</table>
</div>

```

```

<div id="rightcolumn1"><!-- right column -->
<b>your random points</b><br><!-- outputs the set of points the user
assigns to -->
<%
out.println(point1);
%><br>
<%
out.println(point2);
%><br>
<%
out.println(point3);
%><br>
<%
out.println(point4);
%><br>
<%
out.println(point5);
%><br>
<%
out.println(point6);
%>

</div>
</div>
<br>
<div id = "submit">
<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>

```



```


preset.jsp


<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the page for assigning the preset ability scores --&gt;

&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;script src="validRadio.js"&gt;&lt;/script&gt;!-- import 2d radio button functions --&gt;

&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;
&lt;%
session.getAttribute("player5");//latest character object
coursework player = (coursework)session.getAttribute("player5");

int st = 0;//initialise base ability scores
int de = 0;
int co = 0;
int in = 0;
int wi = 0;
int ch = 0;
int check;
boolean[] flag = {false, false, false, false, false, false};//flags to check all
scores are assigned
if (request.getParameter("str") != null) {
    st = Integer.parseInt(request.getParameter("str"));//assign str
    flag[0] = true;//set flag to true
}
if (request.getParameter("dex") != null) {
    de = Integer.parseInt(request.getParameter("dex"));//assign dex
    flag[1] = true;
}
if (request.getParameter("con") != null) {
    co = Integer.parseInt(request.getParameter("con"));//assign con
    flag[2] = true;
}
if (request.getParameter("int") != null) {
    in = Integer.parseInt(request.getParameter("int"));//assign int
    flag[3] = true;
}
if (request.getParameter("wis") != null) {
    wi = Integer.parseInt(request.getParameter("wis"));//assign wis
    flag[4] = true;
}
if (request.getParameter("cha") != null) {
    ch = Integer.parseInt(request.getParameter("cha"));//assign charisma
</pre>

```

```

        flag[5] = true;
    }

check = 0;
for(int x=0; x<6; x++){
    if(flag[x] == true){
        check++;// equals six if all flags are true
    }
    if(check == 6){//all flags true
        player.seta1(st);//add all scores to new values
        player.seta2(de);
        player.seta3(co);
        player.seta4(in);
        player.seta5(wi);
        player.seta6(ch);
        session.setAttribute("player6", player);//save latest object
        response.sendRedirect("class.jsp");//redirect
    }
}
%>
<h1>Bram's character creator</h1>
<h2>
    Preset
</h2>
<form name="abs">
<div id="Leftcolumn1"><!-- left column -->
<b>ability scores:</b><br>
<table>
<tr>
    <td>Strength:</td>
    <td><input type="radio" id = "one1" name = "str" value="15"
onclick = "oneWipe(1);">15 </td> <!-- similar to rand but all values are preset -->
    <td><input type="radio" id = "two1" name = "str" value="14"
onclick = "twoWipe(1);">14 </td> <!-- the wipe functions makes the radio buttons
clear in two dimensions -->
    <td><input type="radio" id = "three1" name = "str" value="13"
onclick = "threeWipe(1);">13 </td><!-- allows only one choice per row and column --
->
    <td><input type="radio" id = "four1" name = "str" value="12"
onclick = "fourWipe(1);">12 </td>
    <td><input type="radio" id = "five1" name = "str" value="10"
onclick = "fiveWipe(1);">10 </td>
    <td><input type="radio" id = "six1" name = "str" value="8"
onclick = "sixWipe(1);">8 </td>
</tr>
<tr>
    <td>Dexterity: </td>
    <td><input type="radio" id = "one2" name = "dex" value="15"
onclick = "oneWipe(2);">15</td>
    <td><input type="radio" id = "two2" name = "dex" value="14"
onclick = "twoWipe(2);">14</td>
    <td><input type="radio" id = "three2" name = "dex" value="13"
onclick = "threeWipe(2);">13</td>
    <td><input type="radio" id = "four2" name = "dex" value="12"
onclick = "fourWipe(2);">12</td>
    <td><input type="radio" id = "five2" name = "dex" value="10"
onclick = "fiveWipe(2);">10</td>

```

```

        <td><input type="radio" id ="six2" name = "dex" value="8"
onclick = "sixWipe(2);">8</td>
    </tr>
    <tr>
        <td>Constitution:</td>
        <td><input type="radio" id ="one3" name = "con" value="15"
onclick = "oneWipe(3);">15</td>
            <td><input type="radio" id ="two3" name = "con" value="14"
onclick = "twoWipe(3);">14</td>
                <td><input type="radio" id ="three3" name = "con" value="13"
onclick = "threeWipe(3);">13</td>
                    <td><input type="radio" id ="four3" name = "con" value="12"
onclick = "fourWipe(3);">12</td>
                        <td><input type="radio" id ="five3" name = "con" value="10"
onclick = "fiveWipe(3);">10</td>
                            <td><input type="radio" id ="six3" name = "con" value="8"
onclick = "sixWipe(3);">8</td>
                        </tr>
                        <tr>
                            <td>Intelligence:</td>
                            <td><input type="radio" id ="one4" name = "int" value="15"
onclick = "oneWipe(4);">15</td>
                                <td><input type="radio" id ="two4" name = "int" value="14"
onclick = "twoWipe(4);">14</td>
                                    <td><input type="radio" id ="three4" name = "int" value="13"
onclick = "threeWipe(4);">13</td>
                                        <td><input type="radio" id ="four4" name = "int" value="12"
onclick = "fourWipe(4);">12</td>
                                            <td><input type="radio" id ="five4" name = "int" value="10"
onclick = "fiveWipe(4);">10</td>
                                                <td><input type="radio" id ="six4" name = "int" value="8"
onclick = "sixWipe(4);">8</td>
                            </tr>
                            <tr>
                                <td>Wisdom:</td>
                                <td><input type="radio" id ="one5" name = "wis" value="15"
onclick = "oneWipe(5);">15</td>
                                    <td><input type="radio" id ="two5" name = "wis" value="14"
onclick = "twoWipe(5);">14</td>
                                        <td><input type="radio" id ="three5" name = "wis" value="13"
onclick = "threeWipe(5);">13</td>
                                            <td><input type="radio" id ="four5" name = "wis" value="12"
onclick = "fourWipe(5);">12</td>
                                                <td><input type="radio" id ="five5" name = "wis" value="10"
onclick = "fiveWipe(5);">10</td>
                                                    <td><input type="radio" id ="six5" name = "wis" value="8"
onclick = "sixWipe(5);">8</td>
                            </tr>
                            <tr>
                                <td>Charisma:</td>
                                <td><input type="radio" id ="one6" name = "cha" value="15"
onclick = "oneWipe(6);">15</td>
                                    <td><input type="radio" id ="two6" name = "cha" value="14"
onclick = "twoWipe(6);">14</td>
                                        <td><input type="radio" id ="three6" name = "cha" value="13"
onclick = "threeWipe(6);">13</td>
                                            <td><input type="radio" id ="four6" name = "cha" value="12"
onclick = "fourWipe(6);">12</td>

```

```
        <td><input type="radio" id ="five6" name = "cha" value="10"
onclick = "fiveWipe(6);">10</td>
        <td><input type="radio" id ="six6" name = "cha" value="8"
onclick = "sixWipe(6);">8</td>
    </tr>
</table>
</div>

<div id="rightcolumn1"><!-- right column -->
<b>your points</b><br><!-- standard scores -->
15
<br>
14
<br>
13
<br>
12
<br>
10
<br>
8

</div>

<div id = "submit">
<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>
```

```

pointBought.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the page for choosing ability scores with points -->

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">

<script>
var S = 0;//holds the cost of the chosen strength
var D = 0;//holds the cost of the chosen dexterity
var Co = 0;//holds the cost of the chosen constitution
var I = 0;//holds the cost of the chosen intelligence
var W = 0;//holds the cost of the chosen wisdom
var Ca = 0;//holds the cost of the chosen charisma
var points = 27 - S - D - Co - I - W - Ca;//calculates how many points the user
has spent when they start with 27
function display(){
    document.getElementById("points").innerHTML = points;//sets the div to show
the remaining points
}
function point1(x, y){
    S = x;//set value of strength cost
    points = 27 - S - D - Co - I - W - Ca;//recalculate points
    var elem = document.getElementById("p1");
    elem.value = points;//save the points value in the invisible radio buttons
    var score = document.getElementById("p2");
    score.value = y;//save the chosen strength score in the invisible radio
buttons
    document.getElementById("points").innerHTML = points;//alter points
remaining display
}
function point2(x, y){
    D = x;//set value of dexterity cost
    points = 27 - S - D - Co - I - W - Ca;//reacalculate
    var elem = document.getElementById("p1");
    elem.value = points;//save points
    var score = document.getElementById("p3");
    score.value = y;//save chosen dex score
    document.getElementById("points").innerHTML = points;
}
function point3(x, y){//same for constitution
    Co = x;
    points = 27 - S - D - Co - I - W - Ca;
    var elem = document.getElementById("p1");
    elem.value = points;
    var score = document.getElementById("p4");
    score.value = y;
    document.getElementById("points").innerHTML = points;
}
function point4(x, y){//same for intelligence
    I = x;
    points = 27 - S - D - Co - I - W - Ca;
    var elem = document.getElementById("p1");
    elem.value = points;
}

```

```

        var score = document.getElementById("p5");
        score.value = y;
        document.getElementById("points").innerHTML = points;
    }
    function point5(x, y){//same for wisdom
        W = x;
        points = 27 - S - D - Co - I - W - Ca;
        var elem = document.getElementById("p1");
        elem.value = points;
        var score = document.getElementById("p6");
        score.value = y;
        document.getElementById("points").innerHTML = points;
    }
    function point6(x, y){//same for charisma
        Ca = x;
        points = 27 - S - D - Co - I - W - Ca;
        var elem = document.getElementById("p1");
        elem.value = points;
        var score = document.getElementById("p7");
        score.value = y;
        document.getElementById("points").innerHTML = points;
    }

```

</script>

```

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body onload = "display();"><!-- display current points -->
<%
session.getAttribute("player5");
coursework player = (coursework)session.getAttribute("player5");//get latest
object
int p = 0;//server side variable for points
int st = 0;//variables for scores
int de = 0;
int co = 0;
int in = 0;
int wi = 0;
int ch = 0;
if (request.getParameter("str") != null && request.getParameter("dex") != null
    && request.getParameter("con") != null && request.getParameter("int") !=
null
    && request.getParameter("wis") != null && request.getParameter("cha") != null)
{ //check the values in the invisible radio buttons
    p = Integer.parseInt(request.getParameter("points"));//get points
    st = Integer.parseInt(request.getParameter("strV"));//get all ability score
values
    de = Integer.parseInt(request.getParameter("dexV"));
    co = Integer.parseInt(request.getParameter("conV"));
    in = Integer.parseInt(request.getParameter("intV"));
    wi = Integer.parseInt(request.getParameter("wisV"));
    ch = Integer.parseInt(request.getParameter("chaV"));
    if(p == 0){//if all points have been spent
        player.seta1(st);//add the chosen values to the repetitive abilities
        player.seta2(de);
        player.seta3(co);
        player.seta4(in);
        player.seta5(wi);
        player.seta6(ch);
    }
}

```

```

        session.setAttribute("player6", player); //save latest object
        response.sendRedirect("class.jsp"); //redirect
    }
    if(p > 0){ //not all points spent
        out.println("you still have points left"); //output not all points
    spent
    }
    if(p < 0){ //too many points spent
        out.println("you have spent too many points"); //output too many
    points spent
    }
}

%>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>

<h1>Bram's character creator</h1>
<h2>
    Point bought
</h2>
<form name="abs">
    <div id="leftcolumn1">
        <b>ability scores:</b><br>
        <table><!-- table for formatting -->
            <tr>
                <td>Strength:</td>
                <td><input type="radio" id = "one1" name = "str" value="0"
onclick = "point1(0, 8);">8 </td> <!-- all options for ability scores -->
                <td><input type="radio" id = "two1" name = "str"
value="1" onclick = "point1(1, 9);">9 </td> <!-- functions alter remaining points -
->
                <td><input type="radio" id = "three1" name = "str"
value="2" onclick = "point1(2, 10);">10 </td>
                <td><input type="radio" id = "four1" name = "str"
value="3" onclick = "point1(3, 11);">11 </td>
                <td><input type="radio" id = "five1" name = "str"
value="4" onclick = "point1(4, 12);">12 </td>
                <td><input type="radio" id = "six1" name = "str"
value="5" onclick = "point1(5, 13);">13 </td>
                <td><input type="radio" id = "five1" name = "str"
value="6" onclick = "point1(7, 14);">14 </td>
                <td><input type="radio" id = "six1" name = "str"
value="7" onclick = "point1(9, 15);">15 </td>
            </tr>
            <tr>
                <td>Dexterity: </td>
                <td><input type="radio" id = "one1" name = "dex"
value="0" onclick = "point2(0, 8);">8 </td>
                <td><input type="radio" id = "two1" name = "dex"
value="1" onclick = "point2(1, 9);">9 </td>
                <td><input type="radio" id = "three1" name = "dex"
value="2" onclick = "point2(2, 10);">10 </td>
            </tr>
    </table>
</div>

```

```

        <td><input type="radio" id = "four1" name = "dex"
value="3"onclick = "point2(3, 11);">11 </td>
        <td><input type="radio" id = "five1" name = "dex"
value="4"onclick = "point2(4, 12);">12 </td>
        <td><input type="radio" id = "six1" name = "dex"
value="5"onclick = "point2(5, 13);">13 </td>
        <td><input type="radio" id = "five1" name = "dex"
value="6"onclick = "point2(7, 14);">14 </td>
        <td><input type="radio" id = "six1" name = "dex"
value="7"onclick = "point2(9, 15);">15 </td>
    </tr>
<tr>
    <td>Constitution:</td>
    <td><input type="radio" id = "one1" name = "con"
value="0"onclick = "point3(0, 8);">8 </td>
    <td><input type="radio" id = "two1" name = "con"
value="1"onclick = "point3(1, 9);">9 </td>
    <td><input type="radio" id = "three1" name = "con"
value="2"onclick = "point3(2, 10);">10 </td>
    <td><input type="radio" id = "four1" name = "con"
value="3"onclick = "point3(3, 11);">11 </td>
    <td><input type="radio" id = "five1" name = "con"
value="4"onclick = "point3(4, 12);">12 </td>
    <td><input type="radio" id = "six1" name = "con"
value="5"onclick = "point3(5, 13);">13 </td>
    <td><input type="radio" id = "five1" name = "con"
value="6"onclick = "point3(7, 14);">14 </td>
    <td><input type="radio" id = "six1" name = "con"
value="7"onclick = "point3(9, 15);">15 </td>
</tr>
<tr>
    <td>Intelligence:</td>
    <td><input type="radio" id = "one1" name = "int"
value="0"onclick = "point4(0, 8);">8 </td>
    <td><input type="radio" id = "two1" name = "int"
value="1"onclick = "point4(1, 9);">9 </td>
    <td><input type="radio" id = "three1" name = "int"
value="2"onclick = "point4(2, 10);">10 </td>
    <td><input type="radio" id = "four1" name = "int"
value="3"onclick = "point4(3, 11);">11 </td>
    <td><input type="radio" id = "five1" name = "int"
value="4"onclick = "point4(4, 12);">12 </td>
    <td><input type="radio" id = "six1" name = "int"
value="5"onclick = "point4(5, 13);">13 </td>
    <td><input type="radio" id = "five1" name = "int"
value="6"onclick = "point4(7, 14);">14 </td>
    <td><input type="radio" id = "six1" name = "int"
value="7"onclick = "point4(9, 15);">15 </td>
</tr>
<tr>
    <td>Wisdom:</td>
    <td><input type="radio" id = "one1" name = "wis"
value="0"onclick = "point5(0, 8);">8 </td>
    <td><input type="radio" id = "two1" name = "wis"
value="1"onclick = "point5(1, 9);">9 </td>
    <td><input type="radio" id = "three1" name = "wis"
value="2"onclick = "point5(2, 10);">10 </td>

```

```

        <td><input type="radio" id ="four1" name = "wis"
value="3"onclick = "point5(3, 11);">11 </td>
        <td><input type="radio" id ="five1" name = "wis"
value="4"onclick = "point5(4, 12);">12 </td>
        <td><input type="radio" id ="six1" name = "wis"
value="5"onclick = "point5(5, 13);">13 </td>
        <td><input type="radio" id ="five1" name = "wis"
value="6"onclick = "point5(7, 14);">14 </td>
        <td><input type="radio" id ="six1" name = "wis"
value="7"onclick = "point5(9, 15);">15 </td>
    </tr>
    <tr>
        <td>Charisma:</td>
        <td><input type="radio" id = "one1" name = "cha"
value="0"onclick = "point6(0, 8);">8 </td>
        <td><input type="radio" id = "two1" name = "cha"
value="1"onclick = "point6(1, 9);">9 </td>
        <td><input type="radio" id = "three1" name = "cha"
value="2"onclick = "point6(2, 10);">10 </td>
        <td><input type="radio" id = "four1" name = "cha"
value="3"onclick = "point6(3, 11);">11 </td>
        <td><input type="radio" id = "five1" name = "cha"
value="4"onclick = "point6(4, 12);">12 </td>
        <td><input type="radio" id = "six1" name = "cha"
value="5"onclick = "point6(5, 13);">13 </td>
        <td><input type="radio" id = "five1" name = "cha"
value="6"onclick = "point6(7, 14);">14 </td>
        <td><input type="radio" id = "six1" name = "cha"
value="7"onclick = "point6(9, 15);">15 </td>
    </tr>
</table>

</div>
<div id="rightcolumn1"><!-- right column showing how many points
the cores cost --&gt;
    &lt;b&gt;your points&lt;/b&gt;&lt;br&gt;
    &lt;div id = "points"&gt;&lt;/div&gt;
    &lt;br&gt;
    &lt;center&gt;
        &lt;table&gt;
            &lt;tr&gt;
                &lt;td&gt;Score&lt;/td&gt;
                &lt;td&gt;Cost&lt;/td&gt;
            &lt;/tr&gt;
            &lt;tr&gt;
                &lt;td&gt;8&lt;/td&gt;
                &lt;td&gt;0&lt;/td&gt;
            &lt;/tr&gt;
            &lt;tr&gt;
                &lt;td&gt;9&lt;/td&gt;
                &lt;td&gt;1&lt;/td&gt;
            &lt;/tr&gt;
            &lt;tr&gt;
                &lt;td&gt;10&lt;/td&gt;
                &lt;td&gt;2&lt;/td&gt;
            &lt;/tr&gt;
            &lt;tr&gt;
                &lt;td&gt;11&lt;/td&gt;
</pre>

```

```

<td>3</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>15</td>
<td>9</td>
</tr>
</table>
</center>

</div>
<div id = "hidden"><!-- hidden div to give values to the jsp -->
<input type = "radio" name = "points" id = "p1" value = "27"
checked><!-- passes js points -->
<input type = "radio" name = "strV" id = "p2" value = "0"
checked><!-- passes chosen ability score values -->
<input type = "radio" name = "dexV" id = "p3" value = "0" checked>
<input type = "radio" name = "conV" id = "p4" value = "0" checked>
<input type = "radio" name = "intV" id = "p5" value = "0" checked>
<input type = "radio" name = "wisV" id = "p6" value = "0" checked>
<input type = "radio" name = "chaV" id = "p7" value = "0" checked>
</div>
<div id = "submit">
<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>

```

## validRadio.js

```
//radio buttons for spending points in the point bought webpage

function oneWipe(id){//treat all the radio buttons with an id starting with one as
a group of radio buttons
    var row = id;//get the wanted value to remain checked
    document.getElementById("one1").checked = false;//clear all checks in the
column
    document.getElementById("one2").checked = false;
    document.getElementById("one3").checked = false;
    document.getElementById("one4").checked = false;
    document.getElementById("one5").checked = false;
    document.getElementById("one6").checked = false;
    switch(row){//check the selected rows
        case 1:
            document.getElementById("one1").checked = true;//if the first one
was checked, re-check it after clearing all
            break;
        case 2:
            document.getElementById("one2").checked = true;
            break;
        case 3:
            document.getElementById("one3").checked = true;
            break;
        case 4:
            document.getElementById("one4").checked = true;
            break;
        case 5:
            document.getElementById("one5").checked = true;
            break;
        case 6:
            document.getElementById("one6").checked = true;
            break;
    }
}
function twoWipe(id){//the same as oneWipe for the second column
    var row = id;
    document.getElementById("two1").checked = false;
    document.getElementById("two2").checked = false;
    document.getElementById("two3").checked = false;
    document.getElementById("two4").checked = false;
    document.getElementById("two5").checked = false;
    document.getElementById("two6").checked = false;
    switch(row){
        case 1:
            document.getElementById("two1").checked = true;
            break;
        case 2:
            document.getElementById("two2").checked = true;
            break;
        case 3:
            document.getElementById("two3").checked = true;
            break;
        case 4:
            document.getElementById("two4").checked = true;
            break;
        case 5:
            document.getElementById("two5").checked = true;
```

```

        break;
    case 6:
        document.getElementById("two6").checked = true;
        break;
    }
}

function threeWipe(id){//third column
    var row = id;
    document.getElementById("three1").checked = false;
    document.getElementById("three2").checked = false;
    document.getElementById("three3").checked = false;
    document.getElementById("three4").checked = false;
    document.getElementById("three5").checked = false;
    document.getElementById("three6").checked = false;
    switch(row){
        case 1:
            document.getElementById("three1").checked = true;
            break;
        case 2:
            document.getElementById("three2").checked = true;
            break;
        case 3:
            document.getElementById("three3").checked = true;
            break;
        case 4:
            document.getElementById("three4").checked = true;
            break;
        case 5:
            document.getElementById("three5").checked = true;
            break;
        case 6:
            document.getElementById("three6").checked = true;
            break;
    }
}

function fourWipe(id){//fourth column
    var row = id;
    document.getElementById("four1").checked = false;
    document.getElementById("four2").checked = false;
    document.getElementById("four3").checked = false;
    document.getElementById("four4").checked = false;
    document.getElementById("four5").checked = false;
    document.getElementById("four6").checked = false;
    switch(row){
        case 1:
            document.getElementById("four1").checked = true;
            break;
        case 2:
            document.getElementById("four2").checked = true;
            break;
        case 3:
            document.getElementById("four3").checked = true;
            break;
        case 4:
            document.getElementById("four4").checked = true;
            break;
        case 5:
            document.getElementById("four5").checked = true;
    }
}

```

```

        break;
    case 6:
        document.getElementById("four6").checked = true;
        break;
    }
}

function fiveWipe(id){//fifth column
    var row = id;
    document.getElementById("five1").checked = false;
    document.getElementById("five2").checked = false;
    document.getElementById("five3").checked = false;
    document.getElementById("five4").checked = false;
    document.getElementById("five5").checked = false;
    document.getElementById("five6").checked = false;
    switch(row){
        case 1:
            document.getElementById("five1").checked = true;
            break;
        case 2:
            document.getElementById("five2").checked = true;
            break;
        case 3:
            document.getElementById("five3").checked = true;
            break;
        case 4:
            document.getElementById("five4").checked = true;
            break;
        case 5:
            document.getElementById("five5").checked = true;
            break;
        case 6:
            document.getElementById("five6").checked = true;
            break;
    }
}

function sixWipe(id){//sixth column
    var row = id;
    document.getElementById("six1").checked = false;
    document.getElementById("six2").checked = false;
    document.getElementById("six3").checked = false;
    document.getElementById("six4").checked = false;
    document.getElementById("six5").checked = false;
    document.getElementById("six6").checked = false;
    switch(row){
        case 1:
            document.getElementById("six1").checked = true;
            break;
        case 2:
            document.getElementById("six2").checked = true;
            break;
        case 3:
            document.getElementById("six3").checked = true;
            break;
        case 4:
            document.getElementById("six4").checked = true;
            break;
        case 5:
            document.getElementById("six5").checked = true;
            break;
    }
}

```

```
        break;
case 6:
    document.getElementById("six6").checked = true;
    break;
}

}
```

## Class selection

### class.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!–the class information and selection page -->
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<script>
function barb(){//display barbarian class info
    document.getElementById("classDiv").innerHTML = "";//clear class div
    document.getElementById("classDiv").innerHTML = "<b>class:</b>
Barbarian";//fill with info
    document.getElementById("description").innerHTML = "";//clear
    document.getElementById("description").innerHTML = "<b>Description:</b> A
fierce warrior of primitive background who can enter +
    "a battle rage";//description of barbarian
    document.getElementById("hitDie").innerHTML = "";//clear
    document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d12";//add
info
    document.getElementById("ability").innerHTML = "";//etc
    document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Strength";
    document.getElementById("saveThrow").innerHTML = "";
    document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Strength & Constitution";
    document.getElementById("proficiencies").innerHTML = "";
    document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon
Proficiencies:</b> Light and medium armour, shields,"+
    "simple and martial weapons";
}
function bard(){//display bard class info
    document.getElementById("classDiv").innerHTML = "";
    document.getElementById("classDiv").innerHTML = "<b>class:</b> Bard";
    document.getElementById("description").innerHTML = "";
    document.getElementById("description").innerHTML = "<b>Description:</b> An
inspiring magician whose power echoes the music "+"
    "of creation";
    document.getElementById("hitDie").innerHTML = "";
    document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d8";
    document.getElementById("ability").innerHTML = "";
    document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Charisma";
    document.getElementById("saveThrow").innerHTML = "";
    document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Dexterity & Charisma";
    document.getElementById("proficiencies").innerHTML = "";
    document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon
Proficiencies:</b> Light armour, simple weapons,"+
    "hand crossbows, longswords, rapiers, shortswords";
}
function cler(){//display cleric class info
    document.getElementById("classDiv").innerHTML = "";
    document.getElementById("classDiv").innerHTML = "<b>class:</b> Cleric";
    document.getElementById("description").innerHTML = "";
```

```

        document.getElementById("description").innerHTML = "<b>Description:</b> A
priestly champion who wields divine magic in"+
        "service of a higher power";
        document.getElementById("hitDie").innerHTML = "";
        document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d8";
        document.getElementById("ability").innerHTML = "";
        document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Wisdom";
        document.getElementById("saveThrow").innerHTML = "";
        document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Wisdom & Charisma";
        document.getElementById("proficiencies").innerHTML = "";
        document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon
Proficiencies:</b> Light and medium armour,"+
        "shields, simple weapons";
    }
    function druid(){//display druid class info
        document.getElementById("classDiv").innerHTML = "";
        document.getElementById("classDiv").innerHTML = "<b>class:</b> Druid";
        document.getElementById("description").innerHTML = "";
        document.getElementById("description").innerHTML = "<b>Description:</b> A
priest of the Old Faith, wielding the power of " +
        "nature - moonlight and plant growth, fire and lightning - and adopting
animal forms";
        document.getElementById("hitDie").innerHTML = "";
        document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d8";
        document.getElementById("ability").innerHTML = "";
        document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Wisdom";
        document.getElementById("saveThrow").innerHTML = "";
        document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Intelligence & Wisdom";
        document.getElementById("proficiencies").innerHTML = "";
        document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon
Proficiencies:</b> Light and medium armour(nonmetal)"+
        ", shields(nonmetal), clubs, daggers, darts, javelins, maces, quaterstaffs,
scimitars, sickles, slings, spears";
    }
    function fight(){//display fighter class info
        document.getElementById("classDiv").innerHTML = "";
        document.getElementById("classDiv").innerHTML = "<b>class:</b> Fighter";
        document.getElementById("description").innerHTML = "";
        document.getElementById("description").innerHTML = "<b>Description:</b> A
master of martial combat, skilled with a " +
        "variety of weapons and armour";
        document.getElementById("hitDie").innerHTML = "";
        document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d10";
        document.getElementById("ability").innerHTML = "";
        document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Strength or Dexterity";
        document.getElementById("saveThrow").innerHTML = "";
        document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Strength & Constitution";
        document.getElementById("proficiencies").innerHTML = "";
        document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon
Proficiencies:</b> All armour, shields, simple and"+
        " martial weapons";
    }
}

```

```

function monk(){//display monk class info
    document.getElementById("classDiv").innerHTML = "";
    document.getElementById("classDiv").innerHTML = "<b>class:</b> Monk";
    document.getElementById("description").innerHTML = "";
    document.getElementById("description").innerHTML = "<b>Description:</b> An
master of martial arts, harnessing the power of the " +
    "body in pursuit of physical and spiritual perfection";
    document.getElementById("hitDie").innerHTML = "";
    document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d8";
    document.getElementById("ability").innerHTML = "";
    document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Dexterity & Wisdom";
    document.getElementById("saveThrow").innerHTML = "";
    document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Strength & Dexterity";
    document.getElementById("proficiencies").innerHTML = "";
    document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon
Proficiencies:</b> Simple weapons, shortswords";
}
function pala(){//display paladin class info
    document.getElementById("classDiv").innerHTML = "";
    document.getElementById("classDiv").innerHTML = "<b>class:</b> Paladin";
    document.getElementById("description").innerHTML = "";
    document.getElementById("description").innerHTML = "<b>Description:</b> A
holy warrior bound to a sacred oath";
    document.getElementById("hitDie").innerHTML = "";
    document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d10";
    document.getElementById("ability").innerHTML = "";
    document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Strength & Charisma";
    document.getElementById("saveThrow").innerHTML = "";
    document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Wisdom & Charisma";
    document.getElementById("proficiencies").innerHTML = "";
    document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon
Proficiencies:</b> All armour, shields, simple and"+
    " martial weapons";
}
function rang(){//display ranger class info
    document.getElementById("classDiv").innerHTML = "";
    document.getElementById("classDiv").innerHTML = "<b>class:</b> Ranger";
    document.getElementById("description").innerHTML = "";
    document.getElementById("description").innerHTML = "<b>Description:</b> A
Warrior who uses marial prowess and nature magic" +
    " to combat threats on the edges of civilization";
    document.getElementById("hitDie").innerHTML = "";
    document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d10";
    document.getElementById("ability").innerHTML = "";
    document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Dexterity & Wisdom";
    document.getElementById("saveThrow").innerHTML = "";
    document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Strength & Dexterity";
    document.getElementById("proficiencies").innerHTML = "";
    document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon
Proficiencies:</b> Light and medium armour,"+
    " shields, simple and martial weapons";
}

```

```

function rogu(){//display rogue class info
    document.getElementById("classDiv").innerHTML = "";
    document.getElementById("classDiv").innerHTML = "<b>class:</b> Rogue";
    document.getElementById("description").innerHTML = "";
    document.getElementById("description").innerHTML = "<b>Description:</b> A
scoundrel who uses stealth and trickery to " +
    "overcome obstacles and enemies";
    document.getElementById("hitDie").innerHTML = "";
    document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d8";
    document.getElementById("ability").innerHTML = "";
    document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Dexterity";
    document.getElementById("saveThrow").innerHTML = "";
    document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Dexterity & Intelligence";
    document.getElementById("proficiencies").innerHTML = "";
    document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon
Proficiencies:</b> Light armour, simple weapons" +
    ", hand crossbows, longswords, rapiers, shortswords";
}

function sorc(){//display sorceror class info
    document.getElementById("classDiv").innerHTML = "";
    document.getElementById("classDiv").innerHTML = "<b>class:</b> Sorcerer";
    document.getElementById("description").innerHTML = "";
    document.getElementById("description").innerHTML = "<b>Description:</b> A
spellcaster who draws on inherent magic from a gift " +
    "or bloodline";
    document.getElementById("hitDie").innerHTML = "";
    document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d6";
    document.getElementById("ability").innerHTML = "";
    document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Charisma";
    document.getElementById("saveThrow").innerHTML = "";
    document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Charisma & Constitution";
    document.getElementById("proficiencies").innerHTML = "";
    document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon
Proficiencies:</b> Daggers, darts, slings, " +
    "quaterstaffs, light crossbows";
}

function warl(){//display warlock class info
    document.getElementById("classDiv").innerHTML = "";
    document.getElementById("classDiv").innerHTML = "<b>class:</b> Warlock";
    document.getElementById("description").innerHTML = "";
    document.getElementById("description").innerHTML = "<b>Description:</b> A
wielder of magic that is derived from a bargain with" +
    " an extraplanar entity";
    document.getElementById("hitDie").innerHTML = "";
    document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d8";
    document.getElementById("ability").innerHTML = "";
    document.getElementById("ability").innerHTML = "<b>Primary Ability:</b>
Charisma";
    document.getElementById("saveThrow").innerHTML = "";
    document.getElementById("saveThrow").innerHTML = "<b>Saving Throw
Proficiencies:</b> Wisdom & Charisma";
    document.getElementById("proficiencies").innerHTML = "";
}

```

```

        document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon Proficiencies:</b> Light armour, simple weapons";
    }
    function wiza(){//display wizard info
        document.getElementById("classDiv").innerHTML = "";
        document.getElementById("classDiv").innerHTML = "<b>class:</b> Wizard";
        document.getElementById("description").innerHTML = "";
        document.getElementById("description").innerHTML = "<b>Description:</b> A scholarly magic-user capable of manipulating the " +
            "structures of reality";
        document.getElementById("hitDie").innerHTML = "";
        document.getElementById("hitDie").innerHTML = "<b>Hit Die:</b> d6";
        document.getElementById("ability").innerHTML = "";
        document.getElementById("ability").innerHTML = "<b>Primary Ability:</b> Intelligence";
        document.getElementById("saveThrow").innerHTML = "";
        document.getElementById("saveThrow").innerHTML = "<b>Saving Throw Proficiencies:</b> Intelligence & Wisdom";
        document.getElementById("proficiencies").innerHTML = "";
        document.getElementById("proficiencies").innerHTML = "<b>Armour and Weapon Proficiencies:</b> daggers, darts, slings, "+ "light crossbows";
    }
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character Viewer</a></li>
    </ul>
</div>

<h1>Bram's character creator</h1>
<h2>
    Choose a class
</h2>
<form><!-- choose class -->
    <center>
        <input type="radio" name = "class" value="1" onclick =
"barb();">Barbarian<!-- functions populate the later divs with relevant info -->
        <input type="radio" name = "class" value="2" onclick =
"bard();">Bard
        <input type="radio" name = "class" value="3" onclick =
"cler();">Cleric
        <input type="radio" name = "class" value="4" onclick =
"druid();">Druid
        <input type="radio" name = "class" value="5" onclick =
"fight();">Fighter
        <input type="radio" name = "class" value="6" onclick =
"monk();">Monk
        <input type="radio" name = "class" value="7" onclick =
"pala();">Paladin
        <input type="radio" name = "class" value="8" onclick =
"rang();">Ranger

```

```

        <input type="radio" name = "class" value="9" onclick =
"rogu();">Rogue
        <input type="radio" name = "class" value="10" onclick =
"sorc();">Sorceror
        <input type="radio" name = "class" value="11" onclick =
"warl();">Warlock
        <input type="radio" name = "class" value="12" onclick =
"wiza();">Wizard
            <h2>class information:</h2>
            <div id = "classDiv"></div><!-- divs for class info -->
            <div id = "description"></div>
            <div id = "hitDie"></div>
            <div id = "ability"></div>
            <div id = "saveThrow"></div>
            <div id = "proficiencies"></div>
            <input type="submit" id = "fixedbutton"><!-- submit button -->
        </center>
    </form>

<%
int selectedItem = 0;
if (request.getParameter("class") != null) {
    selectedItem = Integer.parseInt(request.getParameter("class")); //get chosen
class
}
switch(selectedItem){//switch depending on chosen class
case 1:
    response.sendRedirect("barbarianjsp.jsp");//barbarian, redirect to
appropriate class page for further class specific choices
    break;
case 2:
    response.sendRedirect("bard.jsp");//bard
    break;
case 3:
    response.sendRedirect("cleric.jsp");//cleric
    break;
case 4:
    response.sendRedirect("druid.jsp");//druid
    break;
case 5:
    response.sendRedirect("fighter.jsp");//fighter
    break;
case 6:
    response.sendRedirect("monk.jsp");//monk
    break;
case 7:
    response.sendRedirect("paladin.jsp");//paladin
    break;
case 8:
    response.sendRedirect("ranger.jsp");//ranger
    break;
case 9:
    response.sendRedirect("rogue.jsp");//rogue
    break;
case 10:
    response.sendRedirect("sorcerer.jsp");//sorceror
    break;
case 11:

```

```
    response.sendRedirect("warlock.jsp");//warlock
    break;
case 12:
    response.sendRedirect("wizard.jsp");//wizard
    break;
}
%>

</body>
</html>
```

## barbarian.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the barbarian choice page for the setter method in the coursework object--&gt;
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;

&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;

&lt;script src="valid.js"&gt;&lt;/script&gt;&lt;!-- import checkbox validation function --&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;%
session.getAttribute("player6");
coursework player = (coursework)session.getAttribute("player6");//get latest
object
int s1 = 0;//initialise all choice variables
int s2 = 0;
String item1 = "";
String item2 = "";
boolean flag1 = false;
boolean flag2 = false;
boolean flag3 = false;
if (request.getParameter("skill") != null) {
    try{
        String[] a = request.getParameterValues("skill");//get two
skill options from the checkbox
        s1 = Integer.parseInt(a[0]);
        s2 = Integer.parseInt(a[1]);
        flag3 = true;
    }catch(Exception e){
    }
}
if (request.getParameter("item1") != null) {
    item1 = request.getParameter("item1");//get chosen item
    flag1 = true;//set flag for continue
}
if (request.getParameter("item2") != null) {
    item2 = request.getParameter("item2");//get second item
    flag2 =true;//set second flag
}

if(flag1 == true &amp;&amp; flag2 == true &amp;&amp; flag3 == true){
    player.barbaian(s1, s2, item1, item2);//set barbarian with
parameters
    player.processingFinal();//runs the method to do all the final
character sheet calculations
    session.setAttribute("player7", player);//set the final object
    response.sendRedirect("sheet.jsp");//redirect to confirmation page
}
%&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;</pre>
```

```

        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Barbarian
</h2>
<form>
<div id = "align">
<h3>Skills:</h3><!-- checkbox for proficient skills -->
<b>choose two skills:</b><br>
<input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,2);">animal handling
<input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,2);">athletics
<input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,2);">intimidation
<input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,2);">nature
<input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,2);">perception
<input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,2);">survival<br>
<br>
<h3>Equipment:</h3><!-- radio buttons for first item option -->
<b>choose a martial weapon:</b><br>
<input type="radio" name="item1" value="Greataxe">Greataxe
<input type="radio" name="item1" value="Battleaxe">Battleaxe
<input type="radio" name="item1" value="Flail">Flail
<input type="radio" name="item1" value="Glaive">Glaive
<input type="radio" name="item1" value="Greatsword">Greatsword
<input type="radio" name="item1" value="Halberd">Halberd
<input type="radio" name="item1" value="Lance">Lance
<input type="radio" name="item1" value="Longsword">Longsword
<input type="radio" name="item1" value="Maul">Maul
<input type="radio" name="item1" value="Morningstar">Morningstar
<input type="radio" name="item1" value="Pike">Pike
<input type="radio" name="item1" value="Rapier">Rapier
<input type="radio" name="item1" value="Scimitar">Scimitar
<input type="radio" name="item1" value="Shortsword">Shortsword
<input type="radio" name="item1" value="Trident">Trident
<input type="radio" name="item1" value="War pick">War pick
<input type="radio" name="item1" value="Warhammer">Warhammer
<input type="radio" name="item1" value="Whip">Whip
<br><br>
<b>choose two handaxes or a simple weapon:</b><br><!-- rdio buttons for
second item option -->
<input type="radio" name="item2" value="Two handaxes">Two handaxes
<input type="radio" name="item2" value="Club">Club
<input type="radio" name="item2" value="Dagger">Dagger
<input type="radio" name="item2" value="Greatclub">Greatclub
<input type="radio" name="item2" value="Javelin">Javelin
<input type="radio" name="item2" value="Light hammer">Light hammer
<input type="radio" name="item2" value="Mace">Mace
<input type="radio" name="item2" value="Quaterstaff">Quaterstaff
<input type="radio" name="item2" value="Sickle">Sickle
<input type="radio" name="item2" value="Spear">Spear

```

```
<input type="radio" name="item2" value="Crossbow, Light">Crossbow, light  
<input type="radio" name="item2" value="Dart">Dart  
<input type="radio" name="item2" value="Shortbow">Shortbow  
<input type="radio" name="item2" value="Sling">Sling  
<br><br>  
<input type="submit" id = "fixedbutton"><!-- submit button -->  
</div>  
</form>  
</body>  
</html>
```

## bard.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the bardchoice page for the setter method in the coursework object--&gt;
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;

&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;

&lt;script src="valid.js"&gt;&lt;/script&gt;<!-- import checkbox validation --&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;%
    session.getAttribute("player6");
    coursework player = (coursework)session.getAttribute("player6");//import latest object
    int s1 = 0;//initialise choice variables
    int s2 = 0;
    int s3 = 0;
    String i1 = "";
    String i2 = "";
    String i3 = "";
    String item1 = "";
    String item2 = "";
    String inst = "";
    boolean flag1 = false;
    boolean flag2 = false;
    boolean flag3 = false;
    boolean flag4 = false;
    boolean flag5 = false;
    if (request.getParameter("skill") != null) {
        String[] a = request.getParameterValues("skill");
        s1 = Integer.parseInt(a[0]);//set 3 skill choices
        s2 = Integer.parseInt(a[1]);
        s3 = Integer.parseInt(a[2]);
        flag1 = true;
    }
    if (request.getParameter("inst") != null) {
        try{
            String[] b = request.getParameterValues("inst");
            i1 = b[0];//set 3 instrument proficiencies
            i2 = b[1];
            i3 = b[2];
            flag2 = true;
        }catch(Exception e){
        }
    }
    if (request.getParameter("item1") != null) {
        item1 = request.getParameter("item1");//set first item choice
        flag3 = true;
    }
    if (request.getParameter("item2") != null) {
        item2 = request.getParameter("item2");//set second item choice
        flag4 =true;
    }
}</pre>
```

```

        }
        if (request.getParameter("instI") != null) {
            inst = request.getParameter("instI");//set instrument the bard gets
            flag5 =true;
    }

        if(flag1 == true && flag2 == true && flag3 == true && flag4 == true &&
flag5 == true){//if all flags for choices are true
            player.bard(i1, i2, i3, s1, s2, s3, item1, item2, inst);//bard
            setter method with choice parameters
            player.processingFinal();//process all choices to finalise sheet
            session.setAttribute("player7", player);//set final object
            response.sendRedirect("sheet.jsp");//redirect to finalisation page
        }
    %>
    <div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Bard
</h2>
<form>
<div id = "align">
<h3>Skills:</h3>
<b>choose three skills:</b><br><!-- skill checkbox with thre options -->
<input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,3);">athletics
<input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,3);">acrobatics
<input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,3);">sleight of hand
<input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,3);">stealth
<input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,3);">arcana
<input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,3);">history
<input type="checkbox" name="skill" value="7" onclick =
"checkboxlimit(skill,3);">investigation
<input type="checkbox" name="skill" value="8" onclick =
"checkboxlimit(skill,3);">nature
<input type="checkbox" name="skill" value="9" onclick =
"checkboxlimit(skill,3);">religion
<input type="checkbox" name="skill" value="10" onclick =
"checkboxlimit(skill,3);">animal handling<br>
<input type="checkbox" name="skill" value="11" onclick =
"checkboxlimit(skill,3);">insight
<input type="checkbox" name="skill" value="12" onclick =
"checkboxlimit(skill,3);">medicine
<input type="checkbox" name="skill" value="13" onclick =
"checkboxlimit(skill,3);">perception
<input type="checkbox" name="skill" value="14" onclick =
"checkboxlimit(skill,3);">survival

```

```

<input type="checkbox" name="skill" value="15" onclick =
"checkboxlimit(skill,3);">deception
<input type="checkbox" name="skill" value="16" onclick =
"checkboxlimit(skill,3);">intimidation
<input type="checkbox" name="skill" value="17" onclick =
"checkboxlimit(skill,3);">performance
<input type="checkbox" name="skill" value="18" onclick =
"checkboxlimit(skill,3);">persuasion
<br><br>
<b>choose three instruments:</b><br><!-- instrument checkbox with three
options -->
<input type="checkbox" name="inst" value="Bagpipes" onclick =
"checkboxlimit(inst,3);">Bagpipes
<input type="checkbox" name="inst" value="Drum" onclick =
"checkboxlimit(inst,3);">Drum
<input type="checkbox" name="inst" value="Dulcimer" onclick =
"checkboxlimit(inst,3);">Dulcimer
<input type="checkbox" name="inst" value="Flute" onclick =
"checkboxlimit(inst,3);">Flute
<input type="checkbox" name="inst" value="Lute" onclick =
"checkboxlimit(inst,3);">Lute
<input type="checkbox" name="inst" value="Lyre" onclick =
"checkboxlimit(inst,3);">Lyre
<input type="checkbox" name="inst" value="Horn" onclick =
"checkboxlimit(inst,3);">Horn
<input type="checkbox" name="inst" value="Pan Flute" onclick =
"checkboxlimit(inst,3);">Pan Flute
<input type="checkbox" name="inst" value="Shawm" onclick =
"checkboxlimit(inst,3);">Shawm
<input type="checkbox" name="inst" value="Viol" onclick =
"checkboxlimit(inst,3);">Viol<br>
<br><br>
<b>choose a rapier, a longsword or a simple weapon:</b><br><!-- item 1
radio buttons -->
<input type="radio" name="item1" value="Rapier">Rapier
<input type="radio" name="item1" value="Longsword">Longsword
<input type="radio" name="item1" value="Club">Club
<input type="radio" name="item1" value="Dagger">Dagger
<input type="radio" name="item1" value="Greatclub">Greatclub
<input type="radio" name="item1" value="Javelin">Javelin
<input type="radio" name="item1" value="Light hammer">Light hammer
<input type="radio" name="item1" value="Mace">Mace
<input type="radio" name="item1" value="Quaterstaff">Quaterstaff
<input type="radio" name="item1" value="Sickle">Sickle<br>
<input type="radio" name="item1" value="Spear">Spear
<input type="radio" name="item1" value="Crossbow, light">Crossbow, light
<input type="radio" name="item1" value="Dart">Dart
<input type="radio" name="item1" value="Shortbow">Shortbow
<input type="radio" name="item1" value="Sling">Sling
<br><br>
<b>choose a diplomat's pack or entertainer's pack:</b><br><!-- item 2 radio
buttons -->
<input type="radio" name="item2" value="diplomat's pack">diplomat's pack
<input type="radio" name="item2" value="entertainer's pack">entertainer's
pack
<br><br>
<b>choose an instruments:</b><br><!-- instrument item radio button -->
<input type="radio" name="instI" value="Bagpipes">Bagpipes

```

```
<input type="radio" name="instI" value="Drum">Drum
<input type="radio" name="instI" value="Dulcimer">Dulcimer
<input type="radio" name="instI" value="Flute">Flute
<input type="radio" name="instI" value="Lute">Lute
<input type="radio" name="instI" value="Lyre">Lyre
<input type="radio" name="instI" value="Horn">Horn
<input type="radio" name="instI" value="Pan Flute">Pan Flute
<input type="radio" name="instI" value="Shawm">Shawm
<input type="radio" name="instI" value="Viol">Viol<br>
<br><br>
<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>
```

```

cleric.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
<!–the cleric choice page for the setter method in the coursework object-->
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<script src="valid.js"></script>
<script>
//the different cleric disciplines have different effects, so each has a function
outputting info
function lig(){//light domain
    document.getElementById("spells").innerHTML = "";//reset and add info
    document.getElementById("spells").innerHTML = " <b>you learn the spells:
</b>burning hands, faerie fire";
    document.getElementById("feature").innerHTML = "";
    document.getElementById("feature").innerHTML = "<b>you gain the feature:
</b>Warding flame: you use light to impose " +
        "disadvantage as a reaction";
    document.getElementById("lang").innerHTML = "";
    document.getElementById("langDiv").innerHTML = "";
    document.getElementById("skill").innerHTML = "";
    document.getElementById("skill").innerHTML = "<p>you gain the light
cantrip</p>";
    document.getElementById("skillDiv").innerHTML = "";
}
function lif(){//life domain
    document.getElementById("spells").innerHTML = "";//reset and add info
    document.getElementById("spells").innerHTML = " <b>you learn the spells:
</b>bless, cure wounds";
    document.getElementById("feature").innerHTML = "";
    document.getElementById("feature").innerHTML = "<b>you gain the feature:
</b>Disciple of life: your healing spells are " +
        "more effective";
    document.getElementById("lang").innerHTML = "";
    document.getElementById("langDiv").innerHTML = "";
    document.getElementById("skill").innerHTML = "";
    document.getElementById("skill").innerHTML = "<p>you gain proficiency in
heavy armour</p>";
    document.getElementById("skillDiv").innerHTML = "";
}
function tem(){//tempest
    document.getElementById("spells").innerHTML = "";//reset and add info
    document.getElementById("spells").innerHTML = " <b>you learn the spells:
</b>fog cloud, thunderwave";
    document.getElementById("feature").innerHTML = "";
    document.getElementById("feature").innerHTML = "<b>you gain the feature:
</b>Wrath of the storm: use your reaction to an" +
        "attack to cause thunder damage";
    document.getElementById("lang").innerHTML = "";
    document.getElementById("langDiv").innerHTML = "";
    document.getElementById("skill").innerHTML = "";
    document.getElementById("skill").innerHTML = "<p>you gain proficiency in
heavy armour and martial weapons</p>";
    document.getElementById("skillDiv").innerHTML = "";
}

```

```

}

function tri(){//trickster
    document.getElementById("spells").innerHTML = "";//reset and add info
    document.getElementById("spells").innerHTML = " <b>you learn the spells:  

</b>charm person, disguise self";
    document.getElementById("feature").innerHTML = "";
    document.getElementById("feature").innerHTML = "<b>you gain the feature:  

</b>Blessing of the trickster: give advantage" +
    " to stealth checks";
    document.getElementById("lang").innerHTML = "";
    document.getElementById("langDiv").innerHTML = "";
    document.getElementById("skill").innerHTML = "";
    document.getElementById("skillDiv").innerHTML = "";
}

function war(){//war
    document.getElementById("spells").innerHTML = "";//reset and add info
    document.getElementById("spells").innerHTML = " <b>you learn the spells:  

</b>divine favor, shield of faith";
    document.getElementById("feature").innerHTML = "";
    document.getElementById("feature").innerHTML = "<b>you gain the feature:  

</b>War priest, use your bonus action after" +
    "an attack to attack again";
    document.getElementById("lang").innerHTML = "";
    document.getElementById("langDiv").innerHTML = "";
    document.getElementById("skill").innerHTML = "";
    document.getElementById("skill").innerHTML = "<p>you gain proficiency in  

heavy armour and martial weapons</p>";
    document.getElementById("skillDiv").innerHTML = "";
}

function kno(){//knowledge domain
    document.getElementById("spells").innerHTML = "";//reset and add info
    document.getElementById("spells").innerHTML = " <b>you learn the spells:  

</b>Command, Identify";
    document.getElementById("feature").innerHTML = "";
    document.getElementById("feature").innerHTML = "<b>you gain the feature:  

</b>Blessings of knowledge: you learn two new languages " +
    "and become proficient in two new skills, their proficiency bonus is  

doubled.";
    document.getElementById("lang").innerHTML = "";
    document.getElementById("lang").innerHTML = "<p>you learn two  

languages</p>";
    document.getElementById("langDiv").innerHTML = "";
    document.getElementById("skill").innerHTML = "";
    document.getElementById("skill").innerHTML = "<p>you become proficient in  

two skills</p>";
    document.getElementById("skillDiv").innerHTML = "";

var objDiv = document.getElementById("langDiv");//add radio buttons for extra  

languages

var radioItem1 = document.createElement("input");
radioItem1.type = "checkbox";
radioItem1.name = "langH";
radioItem1.value = "Dwarvish";

var radioItem2 = document.createElement("input");
radioItem2.type = "checkbox";
radioItem2.name = "langH";

```

```

radioItem2.value = "Elvish";

var radioItem3 = document.createElement("input");
radioItem3.type = "checkbox";
radioItem3.name = "langH";
radioItem3.value = "Giant";

var radioItem4 = document.createElement("input");
radioItem4.type = "checkbox";
radioItem4.name = "langH";
radioItem4.value = "Gnomish";

var radioItem5 = document.createElement("input");
radioItem5.type = "checkbox";
radioItem5.name = "langH";
radioItem5.value = "Goblin";

var radioItem6 = document.createElement("input");
radioItem6.type = "checkbox";
radioItem6.name = "langH";
radioItem6.value = "Halfling";

var radioItem7 = document.createElement("input");
radioItem7.type = "checkbox";
radioItem7.name = "langH";
radioItem7.value = "Orc";

var radioItem8 = document.createElement("input");
radioItem8.type = "checkbox";
radioItem8.name = "langH";
radioItem8.value = "Abyssal";

var radioItem9 = document.createElement("input");
radioItem9.type = "checkbox";
radioItem9.name = "langH";
radioItem9.value = "Celestial";

var radioItem10 = document.createElement("input");
radioItem10.type = "checkbox";
radioItem10.name = "langH";
radioItem10.value = "Draconic";

var radioItem11 = document.createElement("input");
radioItem11.type = "checkbox";
radioItem11.name = "langH";
radioItem11.value = "Deep Speech";

var radioItem12 = document.createElement("input");
radioItem12.type = "checkbox";
radioItem12.name = "langH";
radioItem12.value = "Infernal";

var radioItem13 = document.createElement("input");
radioItem13.type = "checkbox";
radioItem13.name = "langH";
radioItem13.value = "Primordial";

```

```

var radioItem14 = document.createElement("input");
radioItem14.type = "checkbox";
radioItem14.name = "langH";
radioItem14.value = "Sylvan";

var radioItem15 = document.createElement("input");
radioItem15.type = "checkbox";
radioItem15.name = "langH";
radioItem15.value = "Undercommon";

var objTextNode1 = document.createTextNode("Dwarvish");//text
var objTextNode2 = document.createTextNode("Elvish");
var objTextNode3 = document.createTextNode("Giant");
var objTextNode4 = document.createTextNode("Gnomish");
var objTextNode5 = document.createTextNode("Goblin");
var objTextNode6 = document.createTextNode("Halfling");
var objTextNode7 = document.createTextNode("Orc");
var objTextNode8 = document.createTextNode("Abyssal");
var objTextNode9 = document.createTextNode("Celestial");
var objTextNode10 = document.createTextNode("Draconic");
var objTextNode11 = document.createTextNode("Deep Speech");
var objTextNode12 = document.createTextNode("Infernal");
var objTextNode13 = document.createTextNode("Primordial");
var objTextNode14 = document.createTextNode("Sylvan");
var objTextNode15 = document.createTextNode("Undercommon");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);

```

```

objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;
objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

var objLabel11 = document.createElement("label");
objLabel11.htmlFor = radioItem11.id;
objLabel11.appendChild(radioItem11);
objLabel11.appendChild(objTextNode11);

var objLabel12 = document.createElement("label");
objLabel12.htmlFor = radioItem12.id;
objLabel12.appendChild(radioItem12);
objLabel12.appendChild(objTextNode12);

var objLabel13 = document.createElement("label");
objLabel13.htmlFor = radioItem13.id;
objLabel13.appendChild(radioItem13);
objLabel13.appendChild(objTextNode13);

var objLabel14 = document.createElement("label");
objLabel14.htmlFor = radioItem14.id;
objLabel14.appendChild(radioItem14);
objLabel14.appendChild(objTextNode14);

var objLabel15 = document.createElement("label");
objLabel15.htmlFor = radioItem15.id;
objLabel15.appendChild(radioItem15);
objLabel15.appendChild(objTextNode15);

objDiv.appendChild(objLabel); //append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);

```

```

objDiv.appendChild(objLabel10);
objDiv.appendChild(objLabel11);
objDiv.appendChild(objLabel12);
objDiv.appendChild(objLabel13);
objDiv.appendChild(objLabel14);
objDiv.appendChild(objLabel15);

var objDiv = document.getElementById("skillDiv");//proficiency radio
buttons
var radioItem1 = document.createElement("input");
radioItem1.type = "checkbox";
radioItem1.name = "profs";
radioItem1.value = "1";

var radioItem2 = document.createElement("input");
radioItem2.type = "checkbox";
radioItem2.name = "profs";
radioItem2.value = "2";

var radioItem3 = document.createElement("input");
radioItem3.type = "checkbox";
radioItem3.name = "profs";
radioItem3.value = "3";

var radioItem4 = document.createElement("input");
radioItem4.type = "checkbox";
radioItem4.name = "profs";
radioItem4.value = "4";

var objTextNode1 = document.createTextNode("arcana");//text
var objTextNode2 = document.createTextNode("history");
var objTextNode3 = document.createTextNode("nature");
var objTextNode4 = document.createTextNode("religion");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

objDiv.appendChild(objLabel);//append

```

```

        objDiv.appendChild(objLabel2);
        objDiv.appendChild(objLabel3);
        objDiv.appendChild(objLabel4);
    }
    function nat(){//nature
        document.getElementById("spells").innerHTML = "";//reset and add info
        document.getElementById("spells").innerHTML = " <b>you learn the spells:  

</b>animal friendship, speak with animals";
        document.getElementById("feature").innerHTML = "";
        document.getElementById("feature").innerHTML = "<b>you gain the feature:  

</b>Acolyte of nature: you learn a druid cantrip " +
        "and gain proficiency in one of the following skills";
        document.getElementById("lang").innerHTML = "";
        document.getElementById("langDiv").innerHTML = "";
        document.getElementById("skill").innerHTML = "";
        document.getElementById("skill").innerHTML = "<p>you become proficient in  

one skill and heavy armour</p>";
        document.getElementById("skillDiv").innerHTML = "";
        var objDiv = document.getElementById("skillDiv");//create proficiency radio
buttons
        var radioItem1 = document.createElement("input");
        radioItem1.type = "radio";
        radioItem1.name = "prof";
        radioItem1.value = "1";

        var radioItem2 = document.createElement("input");
        radioItem2.type = "radio";
        radioItem2.name = "prof";
        radioItem2.value = "2";

        var radioItem3 = document.createElement("input");
        radioItem3.type = "radio";
        radioItem3.name = "prof";
        radioItem3.value = "3";

        var objTextNode1 = document.createTextNode("animal handling");//text
        var objTextNode2 = document.createTextNode("nature");
        var objTextNode3 = document.createTextNode("survival");

        var objLabel = document.createElement("label");
        objLabel.htmlFor = radioItem1.id;
        objLabel.appendChild(radioItem1);
        objLabel.appendChild(objTextNode1);

        var objLabel2 = document.createElement("label");
        objLabel2.htmlFor = radioItem2.id;
        objLabel2.appendChild(radioItem2);
        objLabel2.appendChild(objTextNode2);

        var objLabel3 = document.createElement("label");
        objLabel3.htmlFor = radioItem3.id;
        objLabel3.appendChild(radioItem3);
        objLabel3.appendChild(objTextNode3);

        objDiv.appendChild(objLabel);//append
        objDiv.appendChild(objLabel2);
        objDiv.appendChild(objLabel3);
    }
}

```

```

}
</script>
<script src="valid.js"></script> //import checkbox validation
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<div id="navcontainer">
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<%
    session.getAttribute("player6");
    coursework player = (coursework)session.getAttribute("player6");//get
latest object

    int s1 = 0;//initialise choise
    int s2 = 0;
    boolean flag1 = false;
    if (request.getParameter("skill") != null) {
        try{
            String[] a = request.getParameterValues("skill");//get chosen
skills
            s1 = Integer.parseInt(a[0]);
            s2 = Integer.parseInt(a[1]);
            flag1 = true;
        }catch(Exception e){
        }
    }
    String i1 = "";
    boolean item1 = false;
    boolean flag2 = false;
    if (request.getParameter("item1") != null) {
        i1 = request.getParameter("item1");//get chosen item
        if(i1.equals("1")){
            item1 = true;//setter uses a boolean as there are two choices.
sets to the appropriate boolean value
        }else{
            item1 = false;
        }
        flag2 = true;
    }
    String i2 = "";
    int item2 = 0;
    boolean flag3 = false;
    if (request.getParameter("item2") != null) {
        i2 = request.getParameter("item2");
        item2 = Integer.parseInt(i1);//get chosen item
        flag3 =true;
    }
    String item3 = "";
    boolean flag4 = false;
    if (request.getParameter("item3") != null) {
        item3 = request.getParameter("item3");//get chosen item
        flag4 =true;
    }

```

```

}
String i4 = "";
boolean item4 = false;
boolean flag5 = false;
if (request.getParameter("item4") != null) {
    i4 = request.getParameter("item4");
    if(i4.equals("1")){
        item4 = true;//get chosen item as boolean value
    }else{
        item4 = false;
    }
    flag5 = true;
}
String d = "";
int domain = 0;
boolean flag6 = false;
boolean flag7 = false;
int s3 = 0;
int s4 = 0;
String l1 = "";
String l2 = "";
String p1 = "";
if (request.getParameter("domain") != null) {
    d = request.getParameter("domain");
    domain = Integer.parseInt(d);//get domain numerically
    if(domain == 1){//knowledge
        if (request.getParameter("profs") != null) {
            String[] c = request.getParameterValues("profs");
            s3 = Integer.parseInt(c[0]);//get knowledge proficiency
        }
        attributes
        s4 = Integer.parseInt(c[1]);
        if (request.getParameter("langH") != null) {
            String[] l = request.getParameterValues("langH");
            l1 = l[0];//get knowledge languages
            l2 = l[1];
            player.blessKnowledge(l1, l2, s3, s4);//run bless
        }
        knowledge setter method for knowledge domain
    }
}
if(domain == 4){//nature domain
    if (request.getParameter("prof") != null) {
        s3 =
Integer.parseInt(request.getParameter("prof"));//get chosen skill
        player.acolyteOfNature(s3);//run acolyte of nature
method
    }
}
flag6 = true;//if a domain is chosen, flag is true
}
if(flag1 == true && flag2 == true && flag3 == true && flag4 == true &&
flag5 == true && flag6 == true){//if all choices have been made
try{
    player.cleric(s1, s2, item1, item2, item3, item4,
domain);//run cleric setter with parameters
    player.processingFinal();//process character sheet
    session.setAttribute("player7", player);//set final object
}

```

```

        response.sendRedirect("sheet.jsp");//redirect to finalisation
page
    }catch(Exception e){
        System.out.println(e);
    }
}

%>
<h1>Bram's character creator</h1>
<h2>
    Cleric
</h2>
<form>
<div id = "align">
<h3>Skills:</h3>
<b>choose two skills:</b><br><!-- proficiency checkbox with a limit of two
choices -->
<input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,2);">history
<input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,2);">insight
<input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,2);">medicine
<input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,2);">persuasion
<input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,2);">religion
<br><br>
<h3>Equipment:</h3>
<p>
proficiencies: (<%= player.otherProficiencies %>)<!-- display player
proficiencies so far -->
</p>
<b>choose:</b><br><!-- item 1 radio buttons -->
<input type="radio" name="item1" value="1">Mace <b> or </b>
<input type="radio" name="item1" value="0">Warhammer (if proficient)
<br><br>
<b>and:</b><br><!-- item 2 radio buttons -->
<input type="radio" name="item2" value="1">Scale mail
<input type="radio" name="item2" value="2">Leather armour
<input type="radio" name="item2" value="3">Chain mail (if proficient)
<br>
<br><b>choose a light crossbow and 20 bolts or a simple melee
weapon:</b><br><!-- item 3 radio buttons -->
<input type="radio" name="item3" value="">A light crossbow and 20 bolts
<input type="radio" name="item3" value="Club">Club
<input type="radio" name="item3" value="Dagger">Dagger
<input type="radio" name="item3" value="Greatclub">Greatclub
<input type="radio" name="item3" value="Javelin">Javelin
<input type="radio" name="item3" value="Light hammer">Light hammer
<input type="radio" name="item3" value="Mace">Mace
<input type="radio" name="item3" value="Quaterstaff">Quaterstaff
<input type="radio" name="item3" value="Sickle">Sickle
<input type="radio" name="item3" value="Spear">Spear<br>
<input type="radio" name="item3" value="Crossbow, light">Crossbow, light
<input type="radio" name="item3" value="Dart">Dart
<input type="radio" name="item3" value="Shortbow">Shortbow
<input type="radio" name="item3" value="Sling">Sling

```

```

<br>
<br><b>choose a pack:</b><br> <!-- item 4 radio buttons -->
<input type="radio" name="item4" value="1">A priest's pack <b> or </b>
<input type="radio" name="item4" value="0">An explorers pack
<br><br>
<b>choose a divine domain:</b><br> <!-- domain radio buttons -->
<input type="radio" name="domain" value="1" onclick =
"know();checkboxlimit(langH,2);checkboxlimit(profs,2);">Knowledge
<input type="radio" name="domain" value="2" onclick = "lif();">Life
<input type="radio" name="domain" value="3" onclick = "lig();">Light
<input type="radio" name="domain" value="4" onclick = "nat();">Nature
<input type="radio" name="domain" value="5" onclick = "tem();">Tempest
<input type="radio" name="domain" value="6" onclick = "tri();">Trickery
<input type="radio" name="domain" value="7" onclick = "war();">War
<p><b>domain info:</b></p><!-- information divs -->
<p id = "spells"></p>
<p id = "feature"></p>
<div id = "lang"></div>
<div id = "langDiv"></div>
<div id = "skill"></div>
<div id = "skillDiv"></div>

<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>

```

```

druid.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the druid choice page for the setter method in the coursework object--&gt;

&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;script src="valid.js"&gt;&lt;/script&gt;&lt;!-- import checkbox validation --&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;%
    session.getAttribute("player6");
    coursework player = (coursework)session.getAttribute("player6");//get
latest objec
    int s1 = 0;//initialise choice variables
    int s2 = 0;
    String i1 = "";
    String i2 = "";
    boolean flag1 = false;
    boolean flag2 = false;
    boolean flag3 = false;
    if (request.getParameter("skill") != null) {
        try{
            String[] a = request.getParameterValues("skill");//get two chosen
proficiencies
            s1 = Integer.parseInt(a[0]);
            s2 = Integer.parseInt(a[1]);
            flag1 = true;
            }catch(Exception e){
            }
        }
        if (request.getParameter("item1") != null) {
            i1 = request.getParameter("item1");//get item 1
            flag2 = true;
        }
        if (request.getParameter("item2") != null) {
            i2 = request.getParameter("item2");//get item 2
            flag3 = true;
        }
        if(flag1 == true &amp;&amp; flag2 == true &amp;&amp; flag3 == true){//if all choices are
made
            player.druid(s1, s2, i1, i2);//druid setter with parameters
            player.processingFinal();//final processing method
            session.setAttribute("player7", player);
            response.sendRedirect("sheet.jsp");//redirect to finalisation page
        }
    %&gt;
&lt;div id="navcontainer"&gt;&lt;!-- navigation bar --&gt;
    &lt;ul class="navigation"&gt;
        &lt;li id="active"&gt;&lt;a href="race.jsp"&gt;Character Creator&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="CharacterViewer.jsp" id="current"&gt;Character
Viewer&lt;/a&gt;&lt;/li&gt;
</pre>

```

```

        </ul>
    </div>
<h1>Bram's character creator</h1>
<h2>
    Druid
</h2>
<form>
<div id = "align">
<h3>Skills:</h3>
<b>choose two skills:</b><br><!-- proficiency checkbox choices, must make 2
-->
<input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,2);">arcana
<input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,2);">animal handling
<input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,2);">insight
<input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,2);">medicine
<input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,2);">nature
<input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,2);">perception
<input type="checkbox" name="skill" value="7" onclick =
"checkboxlimit(skill,2);">religion
<input type="checkbox" name="skill" value="8" onclick =
"checkboxlimit(skill,2);">survival<br>
<br>
<h3>Equipment:</h3>
<br><!-- 1st item choice radio buttons -->
<p>choose a wooden shield or simple weapon</p>
<input type="radio" name="item1" value=""> a Wooden shield
<input type="radio" name="item1" value="Club">Club
<input type="radio" name="item1" value="Dagger">Dagger
<input type="radio" name="item1" value="Greatclub">Greatclub
<input type="radio" name="item1" value="Javelin">Javelin
<input type="radio" name="item1" value="Light hammer">Light hammer
<input type="radio" name="item1" value="Mace">Mace
<input type="radio" name="item1" value="Quaterstaff">Quaterstaff
<input type="radio" name="item1" value="Sickle">Sickle
<input type="radio" name="item1" value="Spear">Spear
<input type="radio" name="item1" value="Crossbow, light">Crossbow, light
<input type="radio" name="item1" value="Dart">Dart
<input type="radio" name="item1" value="Shortbow">Shortbow
<input type="radio" name="item1" value="Sling">Sling
<br><!-- second choice radio buttons -->
<p>choose a scimitar or simple melee weapon</p>
<input type="radio" name="item2" value=""> a Scimitar
<input type="radio" name="item2" value="Club">Club
<input type="radio" name="item2" value="Dagger">Dagger
<input type="radio" name="item2" value="Greatclub">Greatclub
<input type="radio" name="item2" value="Javelin">Javelin
<input type="radio" name="item2" value="Light hammer">Light hammer
<input type="radio" name="item2" value="Mace">Mace
<input type="radio" name="item2" value="Quaterstaff">Quaterstaff
<input type="radio" name="item2" value="Sickle">Sickle
<input type="radio" name="item2" value="Spear">Spear
<input type="radio" name="item2" value="Crossbow, light">Crossbow, light

```

```
<input type="radio" name="item2" value="Dart">Dart
<input type="radio" name="item2" value="Shortbow">Shortbow
<input type="radio" name="item2" value="Sling">Sling
<br>
<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>
```

```

fighter.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--the fighter choice page for the setter method in the coursework object-->

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">

<script src="valid.js"></script><!-- import checkbox validation -->
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<%
    session.getAttribute("player6");
    coursework player = (coursework)session.getAttribute("player6");//get latest object
    int s1 = 0;//variables for choices
    int s2 = 0;
    boolean item1 = false;
    String i = "";
    String j = "";
    String k = "";
    String item2 = "";
    String item3 = "";
    boolean item4 = false;
    boolean item5 = false;
    int feature = 0;
    boolean flag1 = false;
    boolean flag2 = false;
    boolean flag3 = false;
    boolean flag4 = false;
    boolean flag5 = false;
    boolean flag6 = false;
    if (request.getParameter("skill") != null) {
        try{
            String[] a = request.getParameterValues("skill");
            s1 = Integer.parseInt(a[0]);//get chosen proficiencies
            s2 = Integer.parseInt(a[1]);
        }catch(Exception e){
        }
    }
    if (request.getParameter("item1") != null) {
        i = request.getParameter("item1");
        if(i.equals("1")){
            item1 = true;//get chosen item as a boolean value
        }else{
            item1 = false;
        }
        flag1 = true;
    }
    if (request.getParameter("item2") != null) {
        item2 = request.getParameter("item2");//get chosen item 2
        flag2 =true;
    }
<%

```

```

        if (request.getParameter("item3") != null) {
            item3 = request.getParameter("item3");//get chosen item 3
            flag3 = true;
        }
        if (request.getParameter("item4") != null) {
            j = request.getParameter("item4");
            if(j.equals("1")){
                item4 = true;//get chosen item 4 as a boolean value
            }else{
                item4 = false;
            }
            flag4 = true;
        }
        if (request.getParameter("item5") != null) {
            k = request.getParameter("item5");
            if(j.equals("1")){
                item5 = true;//get chosen item 5 as a boolean value
            }else{
                item5 = false;
            }
            flag5 = true;
        }
        if (request.getParameter("feat") != null) {
            feature = Integer.parseInt(request.getParameter("feat"));//get
            fighter feature as an integer
            flag6 = true;
        }
        if(flag1 == true && flag2 == true && flag3 == true
            && flag4 == true && flag5 == true && flag6 == true){//if all
            choices are made
            player.fighter(s1, s2, item1, item2, item3, item4, item5,
            feature);//fighter setter with all parameters
            player.processingFinal();//final processing
            session.setAttribute("player7", player);
            response.sendRedirect("sheet.jsp");//redirect to confirmation page
        }
    %>
<div id="navcontainer"><!-- naviagtion bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
        Viewer</a></li>
    </ul>
</div>

<h1>Bram's character creator</h1>
<h2>
    Fighter
</h2>
<form>
    <div id = "align">
        <h3>Skills:</h3>
        <b>choose two skills:</b><br><!-- checkbox for proficiencies, function
        limits to two choices -->
        <input type="checkbox" name="skill" value="1" onclick =
        "checkboxlimit(skill,2);">acrobatics
        <input type="checkbox" name="skill" value="2" onclick =
        "checkboxlimit(skill,2);">animal handling

```

```

<input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,2);">athletics
<input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,2);">history
<input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,2);">insight
<input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,2);">intimidation
<input type="checkbox" name="skill" value="7" onclick =
"checkboxlimit(skill,2);">perception
<input type="checkbox" name="skill" value="8" onclick =
"checkboxlimit(skill,2);">survival<br>
<br>
<div id = "one"></div>
<h3>Equipment:</h3>
<br><!-- item 1 radio buttons -->
<input type="radio" name="item1" value="1">Chainmail <b> or </b>
<input type="radio" name="item1" value="0">Leather, longbow, and 20 arrows
<br>
<b>choose a martial weapon</b><br><!-- item 2 radio buttons -->
<input type="radio" name="item2" value="Battleaxe">Battleaxe
<input type="radio" name="item2" value="Flail">Flail
<input type="radio" name="item2" value="Glaive">Glaive
<input type="radio" name="item2" value="Greataxe">Greataxe
<input type="radio" name="item2" value="Greatsword">Greatsword
<input type="radio" name="item2" value="Halberd">Halberd
<input type="radio" name="item2" value="Lance">Lance
<input type="radio" name="item2" value="Longsword">Longsword
<input type="radio" name="item2" value="Maul">Maul
<input type="radio" name="item2" value="Morningstar">Morningstar
<input type="radio" name="item2" value="Pike">Pike
<input type="radio" name="item2" value="Rapier">Rapier
<input type="radio" name="item2" value="Scimitar">Scimitar
<input type="radio" name="item2" value="Shortsword">Shortsword<br>
<input type="radio" name="item2" value="Trident">Trident
<input type="radio" name="item2" value="War pick">War pick
<input type="radio" name="item2" value="Warhammer">Warhammer
<input type="radio" name="item2" value="Whip">Whip
<input type="radio" name="item2" value="Blowgun">Blowgun
<input type="radio" name="item2" value="Crossbow, hand">Crossbow, hand
<input type="radio" name="item2" value="Crossbow, heavy">Crossbow, heavy
<input type="radio" name="item2" value="Longbow">Longbow
<input type="radio" name="item2" value="Net">Net<br>
<br><b> and another martial weapon or a shield:</b><br><br><!-- item 3 radio
buttons -->
<input type="radio" name="item3" value="">Shield
<input type="radio" name="item3" value="Battleaxe">Battleaxe
<input type="radio" name="item3" value="Flail">Flail
<input type="radio" name="item3" value="Glaive">Glaive
<input type="radio" name="item3" value="Greataxe">Greataxe
<input type="radio" name="item3" value="Greatsword">Greatsword
<input type="radio" name="item3" value="Halberd">Halberd
<input type="radio" name="item3" value="Lance">Lance
<input type="radio" name="item3" value="Longsword">Longsword
<input type="radio" name="item3" value="Maul">Maul
<input type="radio" name="item3" value="Morningstar">Morningstar
<input type="radio" name="item3" value="Pike">Pike
<input type="radio" name="item3" value="Rapier">Rapier

```

```

<input type="radio" name="item3" value="Scimitar">Scimitar
<input type="radio" name="item3" value="Shortsword">Shortsword<br>
<input type="radio" name="item3" value="Trident">Trident
<input type="radio" name="item3" value="War pick">War pick
<input type="radio" name="item3" value="Warhammer">Warhammer
<input type="radio" name="item3" value="Whip">Whip
<input type="radio" name="item3" value="Blowgun">Blowgun
<input type="radio" name="item3" value="Crossbow, hand">Crossbow, hand
<input type="radio" name="item3" value="Crossbow, heavy">Crossbow, heavy
<input type="radio" name="item3" value="Longbow">Longbow
<input type="radio" name="item3" value="Net">Net<br>
<br><b>choose one of the following:</b><br><!-- item 4 radio buttons -->
<input type="radio" name="item4" value="1">A light crossbow and 20 bolts
<b> or </b>
<input type="radio" name="item4" value="0">two handaxes
<br><b>choose a pack:</b><br><!-- item 5 radio button -->
<input type="radio" name="item5" value="1">A dungeoneer's pack <b> or </b>
<input type="radio" name="item5" value="0">An explorers pack
<br><br>
<b>choose a feature:</b><br><!-- feature radio buttons -->
<input type="radio" name="feat" value="1">Archery: +2 to ranged attack
rolls <br>
<input type="radio" name="feat" value="2">Defence: +1 to AC<br>
<input type="radio" name="feat" value="3">Dueling: +2 to one handed melee
attacks ( with no other weapons)<br>
<input type="radio" name="feat" value="4">Great weapon fighting: reroll
1's and 2's for damage with two handed weapons <br>
<input type="radio" name="feat" value="5">Protection: use your shield to
give enemies in 5 ft disadvantage as a reaction <br>
<input type="radio" name="feat" value="6">Two-weapon fighting: add your
ability modifier to weapons used in your off hand <br>
<div id = "two"></div>
<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>

```

### monk.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the monk choice page for the setter method in the coursework object-->

<html>
<head>
<script>

</script>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">

<script src="valid.js"></script><!-- import checkbox validation -->
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>

<%
    session.getAttribute("player6");
    coursework player = (coursework)session.getAttribute("player6");//get
latest object
    boolean flag1 = false;
    boolean flag2 = false;
    boolean flag3 = false;
    boolean flag4 = false;
    int s1 = 0;
    int s2 = 0;
    String i = "";
    String item1 = "";
    String item2 = "";
    if (request.getParameter("i") != null) {
        i = request.getParameter("i");//get chosen item 1
        flag1 = true;
    }
    if (request.getParameter("skill") != null) {
        try{
            String[] a = request.getParameterValues("skill");
            s1 = Integer.parseInt(a[0]);//get chosen proficiencies
            s2 = Integer.parseInt(a[1]);
            flag2 = true;
        }catch(Exception e){
        }
    }
    if (request.getParameter("item1") != null) {
        item1 = request.getParameter("item1");//get item1 choice
        flag3 = true;
    }
    if (request.getParameter("item2") != null) {
        item2 = request.getParameter("item2");//get item2 choice
        flag4 = true;
    }
    if(flag1 == true && flag2 == true && flag3 == true && flag4 == true){
        player.monk(i, s1, s2, item1, item2);//monk setter with choice
parameters
        player.processingFinal();//process character sheet
    }
%>
```

```

        session.setAttribute("player7", player);
        response.sendRedirect("sheet.jsp");
    }
%>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Monk
</h2>
<form>
    <div id = "align">
        <h3>Skills:</h3>
        <p>choose an artisans tool or instrument:</p><!-- i radio buttons -->
        <input type="radio" name="i" value="Alchemist's supplies">alchemist's
supplies
        <input type="radio" name="i" value="Brewer's supplies">Brewer's supplies
        <input type="radio" name="i" value="Calligrapher's
supplies">Calligrapher's supplies
        <input type="radio" name="i" value="Carpenter's tools">Carpenter's tools
        <input type="radio" name="i" value="Carpenter's tools">Carpenter's tools
        <input type="radio" name="i" value="Cartographer's tools">Cartographer's
tools
        <input type="radio" name="i" value="Cobbler's tools">Cobbler's tools
        <input type="radio" name="i" value="Cook's utensil's">Cook's utensil's
        <input type="radio" name="i" value="Glassblower's tools">Glassblower's
tools
        <input type="radio" name="i" value="Jeweler's tools">Jeweler's tools<br>
        <input type="radio" name="i" value="Leatherworker's tools">Leatherworker's
tools
        <input type="radio" name="i" value="Mason's tools">Mason's tools
        <input type="radio" name="i" value="Painter's supplies">Painter's supplies
        <input type="radio" name="i" value="Potter's tools">Potter's tools
        <input type="radio" name="i" value="Smith's tools">Smith's tools
        <input type="radio" name="i" value="Tinker's tools">Tinker's tools
        <input type="radio" name="i" value="Weaver's tools">Weaver's tools
        <input type="radio" name="i" value="Woodcarver's tools">woodcarver's
tools<br>
        <input type="radio" name="i" value="Bagpipes">Bagpipes
        <input type="radio" name="i" value="Drum">Drum
        <input type="radio" name="i" value="Dulcimer">Dulcimer
        <input type="radio" name="i" value="Flute">Flute
        <input type="radio" name="i" value="Lute">Lute
        <input type="radio" name="i" value="Lyre">Lyre
        <input type="radio" name="i" value="Horn">Horn
        <input type="radio" name="i" value="Pan flute">Pan flute
        <input type="radio" name="i" value="Shawm">Shawm
        <input type="radio" name="i" value="Viol">Viol

        <p>choose two skills:</p><!-- proficiency radio buttons, two choices -->
        <input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,2);">acrobatics

```

```

<input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,2);">athletics
<input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,2);">history
<input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,2);">insight
<input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,2);">religion
<input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,2);">stealth<br>
<br>
<h3>Equipment:</h3>

<p>choose a shortsword or simple weapon</p><!-- item 1 radio buttons -->
<input type="radio" name="item1" value="">shortsword
<input type="radio" name="item1" value="Club">Club
<input type="radio" name="item1" value="Dagger">Dagger
<input type="radio" name="item1" value="Greatclub">Greatclub
<input type="radio" name="item1" value="Javelin">Javelin
<input type="radio" name="item1" value="Light hammer">Light hammer
<input type="radio" name="item1" value="Mace">Mace
<input type="radio" name="item1" value="Quaterstaff">Quaterstaff
<input type="radio" name="item1" value="Sickle">Sickle<br>
<input type="radio" name="item1" value="Spear">Spear
<input type="radio" name="item1" value="Crossbow, light">Crossbow, light
<input type="radio" name="item1" value="Dart">Dart
<input type="radio" name="item1" value="Shortbow">Shortbow
<input type="radio" name="item1" value="Sling">Sling

<p>choose a pack:</p><!-- item 2 radio buttons -->
<input type="radio" name="item2" value="1">A dungeoneer's pack <b> or </b>
<input type="radio" name="item2" value="0">An explorers pack
<input type="submit" id = "fixedbutton">
</div>
</form>
</body>
</html>

```

## Paladin.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the paladin choice page for the setter method in the coursework object-->

<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<script src="valid.js"></script><!-- import checkbox validation -->
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<%
session.getAttribute("player6");
coursework player = (coursework)session.getAttribute("player6");//get latest
object
boolean flag1 = false;
boolean flag2 = false;
boolean flag3 = false;
boolean flag4 = false;
boolean flag5 = false;

    int s1 = 0;//initialise choice variables
    int s2 = 0;
    String i1 = "";
    String i2 = "";
    String i3 = "";
    String i4 = "";
    boolean item4 = false;
    if (request.getParameter("skill") != null) {
        try{
            String[] a = request.getParameterValues("skill");
            s1 = Integer.parseInt(a[0]);//get proficvieny choices
            s2 = Integer.parseInt(a[1]);
            flag1 = true;
        }catch(Exception e){
        }
    }
    if (request.getParameter("item1") != null) {
        i1 = request.getParameter("item1");//get item 1 choice
        flag2 = true;
    }
    if (request.getParameter("item2") != null) {
        i2 = request.getParameter("item2");//get item 2 choice
        flag3 = true;
    }
    if (request.getParameter("item3") != null) {
        i3 = request.getParameter("item3");//get item 3 choice
        flag4 = true;
    }
    if (request.getParameter("item4") != null) {
        i4 = request.getParameter("item4");//get item 4 choice
        if(i4.equals("1")){
            item4 = true;//convert to a boolean value
        }else{
```

```

        item4 = false;
    }
    flag5 = true;
}
if(flag1 == true && flag2 == true && flag3 == true && flag4 == true &&
flag5 == true){//if all choices are made
    player.paladin(s1, s2, i1, i2, i3, item4);//set to paladin with
choice parameters
    player.processingFinal();//final processing
    session.setAttribute("player7", player);
    response.sendRedirect("sheet.jsp");//rdirect to final page
}
%>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Paladin
</h2>
<form>
<div id = "align">
<h3>Skills:</h3>

    <p>choose two skills:</p><!-- proficiency checkbox with function limmiting
to two options -->
    <input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,2);">athletics
    <input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,2);">insight
    <input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,2);">intimidation
    <input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,2);">medicine
    <input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,2);">persuasion
    <input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,2);">religion<br>
    <br>
<h3>Equipment:</h3>
<p>choose a martial weapon</p><!-- item 1 radio buttons -->
<input type="radio" name="item1" value="Battleaxe">Battleaxe
<input type="radio" name="item1" value="Flail">Flail
<input type="radio" name="item1" value="Glaive">Glaive
<input type="radio" name="item1" value="Greataxe">Greataxe
<input type="radio" name="item1" value="Greatsword">Greatsword
<input type="radio" name="item1" value="Halberd">Halberd
<input type="radio" name="item1" value="Lance">Lance
<input type="radio" name="item1" value="Longsword">Longsword
<input type="radio" name="item1" value="Maul">Maul
<input type="radio" name="item1" value="Morningstar">Morningstar
<input type="radio" name="item1" value="Pike">Pike
<input type="radio" name="item1" value="Rapier">Rapier
<input type="radio" name="item1" value="Scimitar">Scimitar

```

```

<input type="radio" name="item1" value="Shortsword">Shortsword<br>
<input type="radio" name="item1" value="Trident">Trident
<input type="radio" name="item1" value="War pick">War pick
<input type="radio" name="item1" value="Warhammer">Warhammer
<input type="radio" name="item1" value="Whip">Whip
<p>choose a shield or another martial weapon</p><!-- item 2 radio buttons --
->
<input type="radio" name="item2" value="">Shield
<input type="radio" name="item2" value="Battleaxe">Battleaxe
<input type="radio" name="item2" value="Flail">Flail
<input type="radio" name="item2" value="Glaive">Glaive
<input type="radio" name="item2" value="Greataxe">Greataxe
<input type="radio" name="item2" value="Greatsword">Greatsword
<input type="radio" name="item2" value="Halberd">Halberd
<input type="radio" name="item2" value="Lance">Lance
<input type="radio" name="item2" value="Longsword">Longsword
<input type="radio" name="item2" value="Maul">Maul
<input type="radio" name="item2" value="Morningstar">Morningstar
<input type="radio" name="item2" value="Pike">Pike
<input type="radio" name="item2" value="Rapier">Rapier
<input type="radio" name="item2" value="Scimitar">Scimitar
<input type="radio" name="item2" value="Shortsword">Shortsword<br>
<input type="radio" name="item2" value="Trident">Trident
<input type="radio" name="item2" value="War pick">War pick
<input type="radio" name="item2" value="Warhammer">Warhammer
<input type="radio" name="item2" value="Whip">Whip
<p>choose five javelins or a simple melee weapon</p><!-- item 3 radio
buttons -->
<input type="radio" name="item3" value="">Five javelins
<input type="radio" name="item3" value="Club">Club
<input type="radio" name="item3" value="Dagger">Dagger
<input type="radio" name="item3" value="Greatclub">Greatclub
<input type="radio" name="item3" value="Javelin">Javelin
<input type="radio" name="item3" value="Light hammer">Light hammer
<input type="radio" name="item3" value="Mace">Mace
<input type="radio" name="item3" value="Quaterstaff">Quaterstaff
<input type="radio" name="item3" value="Sickle">Sickle
<input type="radio" name="item3" value="Spear">Spear
<p>choose a pack:</p><!-- item 4 radio buttons -->
<input type="radio" name="item4" value="1">A priest's pack <b> or </b>
<input type="radio" name="item4" value="0">An explorers pack<br><br>
<input type="submit" id = "fixedbutton">
</div>
</form>
</body>
</html>

```

```

ranger.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the ranger choice page for the setter method in the coursework object-->

<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<script>

</script>

<script src="valid.js"></script><!-- import checkbox validation method -->
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<%
    session.getAttribute("player6");
    coursework player = (coursework)session.getAttribute("player6");//get
latest object
    int s1 = 0;//initialise choice variables
    int s2 = 0;
    String i = "";
    boolean item1 = false;
    boolean item2 = false;
    String w1 = "";
    String w2 = "";
    String j = "";
    String favouredEnemy = "";
    boolean pack = false;
    boolean flag1 = false;
    boolean flag2 = false;
    boolean flag3 = false;
    boolean flag4 = false;
    boolean flag5 = false;

    if (request.getParameter("skill") != null) {
        try{
            String[] a = request.getParameterValues("skill");
            s1 = Integer.parseInt(a[0]);//get chosen proficiencies
            s2 = Integer.parseInt(a[1]);
            flag1 = true;
        }catch(Exception e){
        }
    }
    if (request.getParameter("item1") != null) {
        i = request.getParameter("item1");
        if(i.equals("1")){
            item1 = true;//get item 1 as a boolean value
        }else{
            item1 = false;
        }
        flag2 = true;
    }
    if (request.getParameter("item2a") != null) {

```

```

        int s = Integer.parseInt(request.getParameter("item2a"));
        if(s == 1){
            item2 = true;//get item 2a as a boolean value,
            flag3 = true;//either this or the two weapons are needed
        }
    }
    if (request.getParameter("item2") != null) {
        try{
            String[] w = request.getParameterValues("item2");
            w1 = w[0];//get item 2 as two weapons
            w2 = w[1];
            flag3 =true;//these or the shortsword
        }catch(Exception e){
        }
    }
    if (request.getParameter("item3") != null) {
        j = request.getParameter("item3");
        if(j.equals("1")){
            pack = true;//get item  as a boolean value
        }else{
            pack = false;
        }
        flag4 = true;
    }
    if (request.getParameter("enemy") != null) {
        favouredEnemy = request.getParameter("enemy");//get favoured enemy
        as an option
        flag5 = true;
    }
    else{
        if (request.getParameter("enemy1") != null) {
            favouredEnemy += request.getParameter("enemy1");//or get the two
            input humanoids as strings and concatonate them to submit
            favouredEnemy += ", ";
            if (request.getParameter("enemy2") != null) {
                favouredEnemy += request.getParameter("enemy2");
                flag5 = true;
            }
        }
    }
    if(flag1 == true && flag2 == true && flag3 == true&& flag4 == true && flag5
    == true){//if all needed fields are input
        player.ranger(s1, s2, item1, w1, w2, pack, favouredEnemy);//set
        ranger with parameters
        player.processingFinal();//process
        session.setAttribute("player7", player);
        response.sendRedirect("sheet.jsp");//redirect to finalisation page
    }
    %>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
        Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
```

```

Ranger
</h2>
<form>
<div id = "align">
<h3>Skills:</h3>
<b>choose two skills:</b><br><!-- proficiency checkboxes with function
limiting the number of choices to two -->
<input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,2);">animal handling
<input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,2);">athletics
<input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,2);">insight
<input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,2);">investigation
<input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,2);">nature
<input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,2);">perception
<input type="checkbox" name="skill" value="7" onclick =
"checkboxlimit(skill,2);">stealth
<input type="checkbox" name="skill" value="8" onclick =
"checkboxlimit(skill,2);">survival<br>
<br>
<div id = "one"></div>
<h3>Equipment:</h3><!-- item 1 radio buttons -->
<br>
<input type="radio" name="item1" value="1">Scale mail <b> or </b>
<input type="radio" name="item1" value="0">Leather armor
<br>
<p>choose a shortsword or two simple weapons.</p><!-- choose this or two
weapons -->
<input type="checkbox" name="item2a" value="1">two shortswords
<br><b>or</b><br>
<input type="checkbox" name="item2" value="Club" onclick =
"checkboxlimit(item2,2);">Club<!-- two weapons checkbox -->
<input type="checkbox" name="item2" value="Dagger" onclick =
"checkboxlimit(item2,2);">Dagger
<input type="checkbox" name="item2" value="Greatclub" onclick =
"checkboxlimit(item2,2);">Greatclub
<input type="checkbox" name="item2" value="Javelin" onclick =
"checkboxlimit(item2,2);">Javelin
<input type="checkbox" name="item2" value="Light hammer" onclick =
"checkboxlimit(item2,2);">Light hammer
<input type="checkbox" name="item2" value="Mace" onclick =
"checkboxlimit(item2,2);">Mace
<input type="checkbox" name="item2" value="Quaterstaff" onclick =
"checkboxlimit(item2,2);">Quaterstaff
<input type="checkbox" name="item2" value="Sickle" onclick =
"checkboxlimit(item2,2);">Sickle
<input type="checkbox" name="item2" value="Spear" onclick =
"checkboxlimit(item2,2);">Spear
<p>choose a pack:</p><!-- item 3 radio buttons -->
<input type="radio" name="item3" value="1">A dungeoneer's pack <b> or </b>
<input type="radio" name="item3" value="0">An explorers pack
<p>choose a preferred enemy</p><!-- preffered enemy checkbox -->
<input type="radio" name="enemy" value="aberrations">aberrations
<input type="radio" name="enemy" value="beasts">beasts

```

```

<input type="radio" name="enemy" value="celestials">celestials
<input type="radio" name="enemy" value="constructs">constructs
<input type="radio" name="enemy" value="dragons">dragons
<input type="radio" name="enemy" value="elementals">elementals
<input type="radio" name="enemy" value="fey">fey
<input type="radio" name="enemy" value="fiends">fiends
<input type="radio" name="enemy" value="giants">giants
<input type="radio" name="enemy" value="monstrosities">monstrosities
<input type="radio" name="enemy" value="oozes">oozes
<input type="radio" name="enemy" value="plants">plants
<input type="radio" name="enemy" value="undead">undead
<br><b>or two humanoid races: </b>(such as gnolls, orcs or humans)<br><!--
preferred enemy textboxes -->
<input type="text" name="enemy1"> and <input type="text" name="enemy2">
<br><br>
<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>

```

```

rogue.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the rogue choice page for the setter method in the coursework object-->

<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<script src="valid.js"></script><!-- import checkbox validation function -->
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<%
    session.getAttribute("player6");
    coursework player = (coursework)session.getAttribute("player6");//get
latest object
    int s1 = 0;//initialised choice variables
    int s2 = 0;
    int s3 = 0;
    int s4 = 0;
    String i1 = "";
    String i2 = "";
    String i3 = "";
    boolean flag1 = false;
    boolean flag2 = false;
    boolean flag3 = false;
    boolean flag4 = false;
    if (request.getParameter("skill") != null) {
        try{
            String[] a = request.getParameterValues("skill");
            s1 = Integer.parseInt(a[0]);//get four proficiency choices
            s2 = Integer.parseInt(a[1]);
            s3 = Integer.parseInt(a[2]);
            s4 = Integer.parseInt(a[3]);
            flag1 = true;
        }catch(Exception e){
        }
    }
    if (request.getParameter("item1") != null) {
        i1 = request.getParameter("item1");//get item 1
        flag2 = true;
    }
    if (request.getParameter("item2") != null) {
        i2 = request.getParameter("item2");//get item 2
        flag3 =true;
    }
    if (request.getParameter("item3") != null) {
        i3 = request.getParameter("item3");//get item 3
        flag4 =true;
    }

    if(flag1 == true && flag2 == true && flag3 == true
        && flag4 == true){//if all choices are made
        player.rogue(s1, s2, s3, s4, i1, i2, i3);//ste object as rogue with
all parameters
    }

```

```

        session.setAttribute("tempPlayer", player); //set temporary attribute
object
        response.sendRedirect("rogueExpertise.jsp"); //redirect to rogue
specific further choices
    }
    %>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Rogue</h2>
<form>
    <div id = "align">
        <h3>Skills:</h3>
        <b>choose four skills:</b><br><!-- proficiency choices -->
        <input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,4);">acrobatics<!-- limitted to four choices -->
        <input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,4);">athletics
        <input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,4);">deception
        <input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,4);">insight
        <input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,4);">intimidation
        <input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,4);">investigation<br>
        <input type="checkbox" name="skill" value="7" onclick =
"checkboxlimit(skill,4);">perception
        <input type="checkbox" name="skill" value="8" onclick =
"checkboxlimit(skill,4);">performance
        <input type="checkbox" name="skill" value="9" onclick =
"checkboxlimit(skill,4);">persuasion
        <input type="checkbox" name="skill" value="10" onclick =
"checkboxlimit(skill,4);">sleight of hand
        <input type="checkbox" name="skill" value="11" onclick =
"checkboxlimit(skill,4);">stealth
        <br>
        <h3>Equipment:</h3>
        <p>first weapon:</p><!-- item 1 radio buttons -->
        <input type="radio" name="item1" value=", rapier">rapier <b> or </b>
        <input type="radio" name="item1" value=", shortsword">shortsword
        <p>second weapon:</p><!-- item 2 radio buttons -->
        <input type="radio" name="item2" value=", shortbow and quiver of 20
arrows">a shortbow and quiver of 20 arrows <b> or </b>
        <input type="radio" name="item2" value=", shortsword">shortsword
        <p>pack:</p><!-- item 3 radio buttons -->
        <input type="radio" name="item3" value=", burglar's pack">A burglar's pack
<b> or </b>
        <input type="radio" name="item3" value=", dungeoneer's pack">A
dungeoneer's pack <b> or </b>
        <input type="radio" name="item3" value=", explorer's pack">An explorer's
pack

```

```
<br><br>
<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>
```

```

rogueExpertise.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the rogue expertise choice page for choosing which skills to be expert with,
needs to be done after choosing rogue skills, so has its own page-->

<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">

<script src="valid.js"></script><!-- import validation function -->
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<%
    session.getAttribute("tempPlayer");
    coursework player = (coursework)session.getAttribute("tempPlayer");//get latest object
    int e1 = 0;//choice variables inititalised
    int e2 = 0;
    boolean flag1 = false;
    if (request.getParameter("skill") != null) {
        String[] a = request.getParameterValues("skill");
        e1 = Integer.parseInt(a[0]);//get two selected expertises
        e2 = Integer.parseInt(a[1]);
        flag1 = true;
    }
    if(flag1){//if expertises are chosen
        player.expertise(e1, e2);//run expertise method exclusive to rogues
        player.processingFinal();//finalise processing
        session.setAttribute("player7", player);
        response.sendRedirect("sheet.jsp");//redirect to finalise page
    }
%>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Rogue</h2>
<form>
<h3>expertise:</h3>
<div id = "align">
    <b>choose two skills:</b><br><br><!-- user chooses a skill they are already
proficient in -->
    <% if(player.acrobatics){%><input type="checkbox" name="skill"
value="1" onclick = "checkboxlimit(skill,2);">acrobatics<%} %>
    <!-- the if statements only display the option if the user is already
proficient with the skill -->
    <% if(player.animalHandling){%><input type="checkbox" name="skill"
value="2" onclick = "checkboxlimit(skill,2);">animal handling<%} %>

```

```

<!-- any skill the user picks gets its proficiency bonus doubled -->
<% if(player.arcana){%><input type="checkbox" name="skill"
value="3"onclick = "checkboxlimit(skill,2);">arcana<%} %>
<% if(player.athletics){%><input type="checkbox" name="skill"
value="4"onclick = "checkboxlimit(skill,2);">athletics<%} %>
<% if(player.deception){%><input type="checkbox" name="skill"
value="5"onclick = "checkboxlimit(skill,2);">deception<%} %>
<% if(player.history){%><input type="checkbox" name="skill"
value="6"onclick = "checkboxlimit(skill,2);">history<%} %>
<% if(player.insight){%><input type="checkbox" name="skill"
value="7"onclick = "checkboxlimit(skill,2);">insight<%} %>
<% if(player.intimidation){%><input type="checkbox" name="skill"
value="8"onclick = "checkboxlimit(skill,2);">intimidation<%} %>
<% if(player.investigation){%><input type="checkbox" name="skill"
value="9"onclick = "checkboxlimit(skill,2);">investigation<%} %>
<% if(player.medicine){%><input type="checkbox" name="skill"
value="10"onclick = "checkboxlimit(skill,2);">medicine<%} %>
<% if(player.nature){%><input type="checkbox" name="skill"
value="11"onclick = "checkboxlimit(skill,2);">nature<%} %>
<% if(player.perception){%><input type="checkbox" name="skill"
value="12"onclick = "checkboxlimit(skill,2);">perception<%} %>
<% if(player.performance){%><input type="checkbox" name="skill"
value="13"onclick = "checkboxlimit(skill,2);">performance<%} %>
<% if(player.persuasion){%><input type="checkbox" name="skill"
value="14"onclick = "checkboxlimit(skill,2);">persuasion<%} %>
<% if(player.religion){%><input type="checkbox" name="skill"
value="15"onclick = "checkboxlimit(skill,2);">religion<%} %>
<% if(player.sleightOfHand){%><input type="checkbox" name="skill"
value="16"onclick = "checkboxlimit(skill,2);">sleight of hand<%} %>
<% if(player.stealth){%><input type="checkbox" name="skill"
value="17"onclick = "checkboxlimit(skill,2);">stealth<%} %>
<% if(player.survival){%><input type="checkbox" name="skill"
value="18"onclick = "checkboxlimit(skill,2);">survival<%} %>
<br><br>
<input type="submit" id = "fixedbutton"><!-- submit choices -->
</div>
</form>
</body>
</html>

```

```

sorcerer.jsp
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the sorceror choice page for the setter method in the coursework object-->

<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<script>
function clear(){//clears info divs
    document.getElementById("info").innerHTML = "";
    document.getElementById("originDiv").innerHTML = "";
}
function draconic(){//creates draconic ancestry radio buttons
    document.getElementById("info").innerHTML = "";//reset divs and add info
    document.getElementById("info").innerHTML = "Draconic ancestor: <br> choose
one type of dragon as your ancestor";
    document.getElementById("originDiv").innerHTML = "";
    var objDiv = document.getElementById("originDiv");
    var radioItem1 = document.createElement("input");//creates radio buttons
for each type of dragon
    radioItem1.type = "radio";
    radioItem1.name = "dragon";
    radioItem1.value = "1";

    var radioItem2 = document.createElement("input");
    radioItem2.type = "radio";
    radioItem2.name = "dragon";
    radioItem2.value = "2";

    var radioItem3 = document.createElement("input");
    radioItem3.type = "radio";
    radioItem3.name = "dragon";
    radioItem3.value = "3";

    var radioItem4 = document.createElement("input");
    radioItem4.type = "radio";
    radioItem4.name = "dragon";
    radioItem4.value = "4";

    var radioItem5 = document.createElement("input");
    radioItem5.type = "radio";
    radioItem5.name = "dragon";
    radioItem5.value = "5";

    var radioItem6 = document.createElement("input");
    radioItem6.type = "radio";
    radioItem6.name = "dragon";
    radioItem6.value = "6";

    var radioItem7 = document.createElement("input");
    radioItem7.type = "radio";
    radioItem7.name = "dragon";
    radioItem7.value = "7";

    var radioItem8 = document.createElement("input");

```

```

radioItem8.type = "radio";
radioItem8.name = "dragon";
radioItem8.value = "8";

var radioItem9 = document.createElement("input");
radioItem9.type = "radio";
radioItem9.name = "dragon";
radioItem9.value = "9";

var radioItem10 = document.createElement("input");
radioItem10.type = "radio";
radioItem10.name = "dragon";
radioItem10.value = "10";

var objTextNode1 = document.createTextNode("Black");//adds the text saying the
type of dragon
var objTextNode2 = document.createTextNode("Blue");
var objTextNode3 = document.createTextNode("Brass");
var objTextNode4 = document.createTextNode("Bronze");
var objTextNode5 = document.createTextNode("Copper");
var objTextNode6 = document.createTextNode("Gold");
var objTextNode7 = document.createTextNode("Green");
var objTextNode8 = document.createTextNode("Red");
var objTextNode9 = document.createTextNode("Silver");
var objTextNode10 = document.createTextNode("White");

var objLabel = document.createElement("label");
objLabel.htmlFor = radioItem1.id;
objLabel.appendChild(radioItem1);
objLabel.appendChild(objTextNode1);

var objLabel2 = document.createElement("label");
objLabel2.htmlFor = radioItem2.id;
objLabel2.appendChild(radioItem2);
objLabel2.appendChild(objTextNode2);

var objLabel3 = document.createElement("label");
objLabel3.htmlFor = radioItem3.id;
objLabel3.appendChild(radioItem3);
objLabel3.appendChild(objTextNode3);

var objLabel4 = document.createElement("label");
objLabel4.htmlFor = radioItem4.id;
objLabel4.appendChild(radioItem4);
objLabel4.appendChild(objTextNode4);

var objLabel5 = document.createElement("label");
objLabel5.htmlFor = radioItem5.id;
objLabel5.appendChild(radioItem5);
objLabel5.appendChild(objTextNode5);

var objLabel6 = document.createElement("label");
objLabel6.htmlFor = radioItem6.id;
objLabel6.appendChild(radioItem6);
objLabel6.appendChild(objTextNode6);

var objLabel7 = document.createElement("label");
objLabel7.htmlFor = radioItem7.id;

```

```

objLabel7.appendChild(radioItem7);
objLabel7.appendChild(objTextNode7);

var objLabel8 = document.createElement("label");
objLabel8.htmlFor = radioItem8.id;
objLabel8.appendChild(radioItem8);
objLabel8.appendChild(objTextNode8);

var objLabel9 = document.createElement("label");
objLabel9.htmlFor = radioItem9.id;
objLabel9.appendChild(radioItem9);
objLabel9.appendChild(objTextNode9);

var objLabel10 = document.createElement("label");
objLabel10.htmlFor = radioItem10.id;
objLabel10.appendChild(radioItem10);
objLabel10.appendChild(objTextNode10);

objDiv.appendChild(objLabel);//append
objDiv.appendChild(objLabel2);
objDiv.appendChild(objLabel3);
objDiv.appendChild(objLabel4);
objDiv.appendChild(objLabel5);
objDiv.appendChild(objLabel6);
objDiv.appendChild(objLabel7);
objDiv.appendChild(objLabel8);
objDiv.appendChild(objLabel9);
objDiv.appendChild(objLabel10);
}

</script>
<script src="valid.js"></script><!-- import checkbox validation function -->
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<%
session.getAttribute("player6");
coursework player = (coursework)session.getAttribute("player6");//get latest
character object
int s1 = 0;//initialise choice variables
int s2 = 0;
String i1 = "";
String i2 = "";
String i3 = "";
int origin = 0;
boolean flag1 = false;
boolean flag2 = false;
boolean flag3 = false;
boolean flag4 = false;
if (request.getParameter("skill") != null) {
    try{
        String[] a = request.getParameterValues("skill");
        s1 = Integer.parseInt(a[0]);//get chosen skills
        s2 = Integer.parseInt(a[1]);
        flag1 = true;
    }catch(Exception e){
    }
}

```

```

if (request.getParameter("item1") != null) {
    i1 = request.getParameter("item1");//get item 1
    flag2 = true;
}
if (request.getParameter("item2") != null) {
    i2 = request.getParameter("item2");//get item 2
    flag3 =true;
}
if (request.getParameter("item3") != null) {
    i3 = request.getParameter("item3");//get item 3
    flag4 =true;
}
if (request.getParameter("origin") != null) {//gets origin
    if(request.getParameter("origin").equals("0")){
        origin = Integer.parseInt(request.getParameter("origin"));//if
origin is 0, its wild magic
        flag4 =true;
    }else{
        origin = Integer.parseInt(request.getParameter("dragon"));//if
origin is greater than zero it is a type of draconic depending on the number
        flag4 =true;
    }
}

if(flag1 == true && flag2 == true && flag3 == true
    && flag4 == true){//if all choices have been made
    player.sorcerer(s1, s2, i1, i2, i3, origin);//set as sorceror with
choices as parameters
    player.processingFinal();//process
    session.setAttribute("player7", player);
    response.sendRedirect("sheet.jsp");//redirect to final page
}
%>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Sorcerer</h2>
<form>
    <h3>Skills:</h3>
    <div id = "align">
        <b>choose two skills:</b><br><br><!-- proficiency checkbox, limited to two
options with function -->
        <input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,2);">arcana
        <input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,2);">deception
        <input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,2);">insight
        <input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,2);">intimidation
        <input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,2);">persuasion

```

```

<input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,2);">religion<br>
<br>
<h3>Equipment:</h3>
<p>choose a light crossbow and 20 bolts or a simple weapon:</p><!-- item 1
radio buttons -->
<input type="radio" name="item1" value=", A Light crossbow and 20 bolts">A
light crossbow and 20 bolts
<input type="radio" name="item1" value=", Club">Club
<input type="radio" name="item1" value=", Dagger">Dagger
<input type="radio" name="item1" value=", Greatclub">Greatclub
<input type="radio" name="item1" value=", Javelin">Javelin
<input type="radio" name="item1" value=", Light hammer">Light hammer
<input type="radio" name="item1" value=", Mace">Mace
<input type="radio" name="item1" value=", Quaterstaff">Quaterstaff
<input type="radio" name="item1" value=", Sickle">Sickle<br>
<input type="radio" name="item1" value=", Spear">Spear
<input type="radio" name="item1" value=", Crossbow, light">Crossbow, light
<input type="radio" name="item1" value=", Dart">Dart
<input type="radio" name="item1" value=", Shortbow">Shortbow
<input type="radio" name="item1" value=", Sling">Sling
<p>choose a component pouch or an arcane focus:</p><!-- item 2 radio
buttons -->
<input type="radio" name="item2" value=", component pouch">component pouch
<b> or </b>
<input type="radio" name="item2" value=", arcane focus">arcane focus
<p>pack:</p><!-- item 3 radio buttons -->
<input type="radio" name="item3" value=", dungeoneer's pack">dungeoneer's
pack <b> or </b>
<input type="radio" name="item3" value=", explorer's pack">explorer's pack
<p>choose a sorcerous origin:</p><!-- origin radio buttons -->
<input type="radio" name="origin" value="1" onclick =
"draconic();">Draconic bloodline <b> or </b>
<input type="radio" name="origin" value="0" onclick = "clear();">Wild
magic
<br><br>
<div id = "info"></div><!-- divs for outputting origin info and options -->
<div id = "originDiv"></div>
<br>
<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>

```

### warlock.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the warlock choice page for the setter method in the coursework object-->

<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<script src="valid.js"></script><!-- import checkbox validation function -->
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<%
session.getAttribute("player6");
coursework player = (coursework)session.getAttribute("player6");//latest object
int s1 = 0;//initialise choice variables
int s2 = 0;
String item1 = "";
String item2 = "";
String item3 = "";
String item4 = "";
int patron = 0;
boolean flag1 = false;
boolean flag2 = false;
boolean flag3 = false;
boolean flag4 = false;
boolean flag5 = false;
boolean flag6 = false;
if (request.getParameter("skill") != null) {
    try{
        String[] a = request.getParameterValues("skill");
        s1 = Integer.parseInt(a[0]);//get chosen proficiencies
        s2 = Integer.parseInt(a[1]);
        flag1 = true;
    }catch(Exception e){
    }
}
if (request.getParameter("item1") != null) {
    item1 = request.getParameter("item1");//get item 1
    flag2 = true;
}
if (request.getParameter("item2") != null) {
    item2 = request.getParameter("item2");//get item 2
    flag3 =true;
}
if (request.getParameter("item3") != null) {
    item3 = request.getParameter("item3");//get item 3
    flag4 =true;
}
if (request.getParameter("item4") != null) {
    item4 = request.getParameter("item4");//get item 4
    flag5 =true;
}
if (request.getParameter("patron") != null) {
```

```

        patron = Integer.parseInt(request.getParameter("patron")); //get
chosen patron
        flag6 =true;
    }
    if(flag1 == true && flag2 == true && flag3 == true && flag4 == true &&
flag5 == true && flag6 == true){//if all choices are made
        player.warlock(s1,s2,item1,item2,item3,item4,patron); //set as
warlock with chosen parameters
        player.processingFinal(); //process character sheet
        session.setAttribute("player7", player);
        response.sendRedirect("sheet.jsp"); //redirect to finalisation page
    }
%>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Warlock
</h2>
<form>
    <div id = "align">
        <h3>Skills:</h3>
        <b>choose two skills:</b><br><!-- proficiency checkbox with a max of two
choices -->
        <input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,2);">arcana
        <input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,2);">deception
        <input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,2);">history
        <input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,2);">intimidation
        <input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,2);">investigation
        <input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,2);">nature
        <input type="checkbox" name="skill" value="7" onclick =
"checkboxlimit(skill,2);">religion<br>
        <br>
        <h3>Equipment:</h3><!-- item 1 radio buttons -->
        <p>choose a light crossbow and 20 bolts or a simple weapon:</p>
        <input type="radio" name="item1" value=", A Light crossbow and 20 bolts">A
light crossbow and 20 bolts
        <input type="radio" name="item1" value=", Club">Club
        <input type="radio" name="item1" value=", Dagger">Dagger
        <input type="radio" name="item1" value=", Greatclub">Greatclub
        <input type="radio" name="item1" value=", Javelin">Javelin
        <input type="radio" name="item1" value=", Light hammer">Light hammer
        <input type="radio" name="item1" value=", Mace">Mace
        <input type="radio" name="item1" value=", Quaterstaff">Quaterstaff
        <input type="radio" name="item1" value=", Sickle">Sickle
        <input type="radio" name="item1" value=", Spear">Spear
        <input type="radio" name="item1" value=", Crossbow, light">Crossbow, light

```

```

<input type="radio" name="item1" value=", Dart">Dart
<input type="radio" name="item1" value=", Shortbow">Shortbow
<input type="radio" name="item1" value=", Sling">Sling
<p>choose a simple weapon:</p><!-- item 2 radio buttons -->
<input type="radio" name="item2" value=", Club">Club
<input type="radio" name="item2" value=", Dagger">Dagger
<input type="radio" name="item2" value=", Greatclub">Greatclub
<input type="radio" name="item2" value=", Javelin">Javelin
<input type="radio" name="item2" value=", Light hammer">Light hammer
<input type="radio" name="item2" value=", Mace">Mace
<input type="radio" name="item2" value=", Quaterstaff">Quaterstaff
<input type="radio" name="item2" value=", Sickle">Sickle
<input type="radio" name="item2" value=", Spear">Spear
<input type="radio" name="item2" value=", Crossbow, light">Crossbow, light
<input type="radio" name="item2" value=", Dart">Dart
<input type="radio" name="item2" value=", Shortbow">Shortbow
<input type="radio" name="item2" value=", Sling">Sling
<p>choose a component pouch or an arcane focus:</p><!-- item 3 radio
buttons -->
<input type="radio" name="item3" value=", component pouch">component pouch
<b> or </b>
<input type="radio" name="item3" value=", arcane focus">arcane focus
<p>pack:</p><!-- item 4 radio buttons -->
<input type="radio" name="item4" value=", scholar's pack">scholar's pack
<b> or </b>
<input type="radio" name="item4" value=", dungeoneer's pack">dungeoneer's
pack
<p>choose an otherworldly patron:</p><!-- patron radio buttons -->
<input type="radio" name="patron" value="0" onclick = "">Archfey
<input type="radio" name="patron" value="1" onclick = "">Fiend
<input type="radio" name="patron" value="2" onclick = "">Great Old One
<br><br>
<input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>

```

### wizard.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the wizard choice page for the setter method in the coursework object--&gt;

&lt;html&gt;
&lt;head&gt;
&lt;link rel="stylesheet" type="text/css" href="CourseworkApp.css"&gt;
&lt;script src="valid.js"&gt;&lt;/script&gt;&lt;!-- import checkbox validation --&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;
&lt;title&gt;character creator&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;%
session.getAttribute("player6");
coursework player = (coursework)session.getAttribute("player6");//get latest
object
int s1 = 0;//initialise choice variables
int s2 = 0;
String item1 = "";
String item2 = "";
String item3 = "";
boolean flag1 = false;
boolean flag2 = false;
boolean flag3 = false;
boolean flag4 = false;
if (request.getParameter("skill") != null) {
    try{
        String[] a = request.getParameterValues("skill");
        s1 = Integer.parseInt(a[0]);//get chosen proficiencies
        s2 = Integer.parseInt(a[1]);
        flag1 = true;
    }catch(Exception e){
    }
}
if (request.getParameter("item1") != null) {
    item1 = request.getParameter("item1");//get item 1 choice
    flag2 = true;
}
if (request.getParameter("item2") != null) {
    item2 = request.getParameter("item2");//get item 2 choice
    flag3 =true;
}
if (request.getParameter("item3") != null) {
    item3 = request.getParameter("item3");//get item 3 choice
    flag4 =true;
}
if(flag1 == true &amp;&amp; flag2 == true &amp;&amp; flag3 == true &amp;&amp; flag4 == true){//if
all choices are made
    player.wizard(s1, s2, item1, item2, item3);//set player as a wizard
with the chosen parameters
    player.processingFinal();//process sheet
    session.setAttribute("player7", player);
    response.sendRedirect("sheet.jsp");//redirect to the finalisation
page
}</pre>
```

```

%>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<h1>Bram's character creator</h1>
<h2>
    Wizard
</h2>
<form>
<div id = "align">
<h3>Skills:</h3>
<b>choose two skills:</b><br><!-- proficiency checkbox, max of two choices
-->
    <input type="checkbox" name="skill" value="1" onclick =
"checkboxlimit(skill,2);">arcana
    <input type="checkbox" name="skill" value="2" onclick =
"checkboxlimit(skill,2);">history
    <input type="checkbox" name="skill" value="3" onclick =
"checkboxlimit(skill,2);">insight
    <input type="checkbox" name="skill" value="4" onclick =
"checkboxlimit(skill,2);">investigation
    <input type="checkbox" name="skill" value="5" onclick =
"checkboxlimit(skill,2);">medicine
    <input type="checkbox" name="skill" value="6" onclick =
"checkboxlimit(skill,2);">religion<br>
<br>
<h3>Equipment:</h3>
<p>choose a quaterstaff or dagger:</p><!-- item 1 radio buttons -->
    <input type="radio" name="item1" value=", quaterstaff"> quaterstaff
    <input type="radio" name="item1" value=", dagger">dagger
<p>choose a component pouch or an arcane focus:</p><!-- item 2 radio
buttons -->
    <input type="radio" name="item2" value=", component pouch">component pouch
<b> or </b>
    <input type="radio" name="item2" value=", arcane focus">arcane focus
<p>pack:</p><!-- item 3 radio buttons -->
    <input type="radio" name="item3" value=", scholar's pack">scholar's pack
<b> or </b>
    <input type="radio" name="item3" value=", dungeoneer's pack">explorer's
pack
<br><br>
    <input type="submit" id = "fixedbutton"><!-- submit button -->
</div>
</form>
</body>
</html>

```

## finalise

### sheet.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page import="com.google.appengine.api.datastore.DatastoreServiceFactory"%>
    <%@ page import="com.google.appengine.api.datastore.DatastoreService"%>
    <%@ page import="com.google.appengine.api.datastore.Query"%>
    <%@ page import="com.google.appengine.api.datastore.Entity"%>
    <%@ page import="com.google.appengine.api.datastore.FetchOptions"%>
    <%@ page import="com.google.appengine.api.datastore.Key"%>
    <%@ page import="com.google.appengine.api.datastore.KeyFactory"%>
    <%@ page import="com.google.appengine.api.datastore.Query.Filter"%>
    <%@ page import="com.google.appengine.api.datastore.Query.FilterOperator"%>
    <%@ page import="com.google.appengine.api.datastore.Query.FilterPredicate"%>
    <%@ page import="com.google.appengine.api.datastore.Query.CompositeFilter"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%><!-- import app engine classes -->
<!DOCTYPE html>
<!--the character saving page -->

<jsp:useBean id="helper" class="helper.Login" scope="page" /><!-- bean for saving
character to datastore -->
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<style>
h3{
    margin-top:-1%;/*simple alignment*/
}
</style>
<script src="valid.js"></script><!-- import checkbox validation -->
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
<%
    session.getAttribute("player7");
    coursework player = (coursework)session.getAttribute("player7");//get final
character object

    DatastoreService datastore =
DatastoreServiceFactory.getDatastoreService(); //get datastore service
    String username = (String)session.getAttribute("username"); //get username
    %>
<h1>Bram's character creator</h1>
<h2>
    Character
</h2>
<body>
<div id="page1"><!-- left column -->
<h3>Page 1</h3><!-- displays main page information for character sheet -->
```

```

<%
out.println("character name: " + player.pName + "<br>");
out.println("| class: " + player.Class + " background: " + player.background + "
real name: " + player.name+ "<br>");
out.println("| race: " + player.race + " Alignment: " + player.alignment + " xp: "
+ player.xp+ "<br>");
out.println();
out.println("Strength: " + player.Str + " mod: " + player.StrMod + " save: " +
player.strSaveV);
out.println(" |AC: "+player.ac + "|" + " |initiative: "+player.initiative + "|"
+ " |speed: "+player.speed + "|" + "<br>");
out.println("Dexterity: " + player.Dex + " mod: " + player.DexMod + " save: " +
player.dexSaveV+ "<br>");
out.println("Constitution: " + player.Con + " mod: " + player.ConMod + " save:
"+ player.conSaveV);
out.println(" |HP: "+player.hitPoints + "|" + " |hit Dice: "+player.hitDice + |
+ "<br>");
out.println("Intelligence: " + player.Int + " mod: " + player.IntMod + " save:
"+ player.intSaveV+ "<br>");
out.println("Wisdom: " + player.Wis + " mod: " + player.WisMod + " save: " +
player.wisSaveV+ "<br>");
out.println("Charisma: " + player.Cha + " mod: " + player.ChaMod + " save: " +
player.chaSaveV+ "<br><br>");

out.println("personality traits: " + player.personality + " <br>");
out.println("ideals: " + player.ideal + " <br>");
out.println("bonds: " + player.bond + " <br>");
out.println("flaws: " + player.flaw + " <br><br>");

out.println( player.acrobatics + " acrobatics: " + player.acrobaticsV + "<br>");
out.println( player.animalHandling + " animalHandling: " + player.animalHandlingV
+ "<br>");
out.println( player.arcana + " arcana: " + player.arcanaV + "<br>");
out.println( player.athletics + " athletics: " + player.athleticsV + "<br>");
out.println( player.deception + " deception: " + player.deceptionV + "<br>");
out.println( player.history + " history: " + player.historyV + "<br>");
out.println( player.insight + " insight: " + player.insightV + "<br>");
out.println( player.intimidation + " intimidation: " + player.intimidationV +
"<br>");
out.println( player.investigation + " investigation: " + player.investigationV +
"<br>");
out.println( player.medicine + " medicine: " + player.medicineV + "<br>");
out.println( player.nature + " nature: " + player.natureV + "<br>");
out.println( player.perception + " perception: " + player.perceptionV + "<br>");
out.println( player.performance + " performance: " + player.performanceV +
"<br>");
out.println( player.persuasion + " persuasion: " + player.persuasionV + "<br>");
out.println( player.religion + " religion: " + player.religionV + "<br>");
out.println( player.sleightOfHand + " sleightOfHand: " + player.sleightOfHandV +
"<br>");
out.println( player.stealth + " stealth: " + player.stealthV + "<br>");
out.println( player.survival + " survival: " + player.survivalV + "<br>");
out.println("passive perception: " + player.passivePerception + " <br><br>");

out.println("other proficiencies and languages: <br>" + player.languages + "<br>
+ player.otherProficiencies + "<br>");
out.println(" |gold: " + player.money + " | equipment: " + player.equipment + " |
<br>");
```

```

out.println("features and traits: " + player.features);
%>
</div>
<div id="page2"><!-- right column -->
<h3>page 2</h3><!-- displays second page information -->
<%
out.println("character name: " + player.name + "<br>");
out.println("|age: " + player.age + " height: " + player.realH + " weight:" +
player.realW + " | <br>"); 
out.println("|eyes: " + player.eyes + " skin: " + player.skin + " hair:" +
player.hair + " | <br>"); 
out.println("backstory: " + player.backstory);
%>
<%
if (request.getParameter("confirm") != null){//if user saves
    int slot = Integer.parseInt(request.getParameter("confirm")); //get save
slot the user wants to save the character to
    helper.createCharacter(username, player, slot); //use helper to save the
final object to the datastore in that save slot
    response.sendRedirect("CharacterViewer.jsp"); //redirect to character viewer
page
}
%>
</div>
<div id = "center" align="center">
<form>
<h3>Save character</h3>
<input type="radio" name="confirm" value="1" checked> save slot 1<!-- radio
button for choosing which save slot to use -->
<input type="radio" name="confirm" value="2"> save slot 2
<input type="radio" name="confirm" value="3"> save slot 3
<input type="radio" name="confirm" value="4"> save slot 4
<input type="radio" name="confirm" value="5"> save slot 5
<input type="submit" id = "fixedbutton"><!-- submit choice -->
</form>
</div>

</body>
</html>

```

## Character viewer

### characterViewer.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page import="com.google.appengine.api.datastore.DatastoreServiceFactory"%>
<%@ page import="com.google.appengine.api.datastore.DatastoreService"%>
<%@ page import="com.google.appengine.api.datastore.Query"%>
<%@ page import="com.google.appengine.api.datastore.Entity"%>
<%@ page import="com.google.appengine.api.datastore.FetchOptions"%>
<%@ page import="com.google.appengine.api.datastore.Key"%>
<%@ page import="com.google.appengine.api.datastore.KeyFactory"%>
<%@ page import="com.google.appengine.api.datastore.Query.Filter"%>
<%@ page import="com.google.appengine.api.datastore.Query.FilterOperator"%>
<%@ page import="com.google.appengine.api.datastore.Query.FilterPredicate"%>
<%@ page import="com.google.appengine.api.datastore.Query.CompositeFilter"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%><!-- import datastore classes -->
<!DOCTYPE html>
<!--the page for selecting which page of the character sheet to view-->

<jsp:useBean id="helper" class="helper.Login" scope="page" /><!-- -->
<html>
<head>
<link rel="stylesheet" type="text/css" href="CourseworkApp.css">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
</head>
<body>
<%
    String username = (String)session.getAttribute("username");//get username

    if (request.getParameter("save") != null && request.getParameter("sheet")
!= null){
        int save = Integer.parseInt(request.getParameter("save"));
        session.setAttribute("save", save);//set the chosen save slot as a
session attribute, so the print pages output the correct character slot
        if(request.getParameter("sheet").equals("1")){
            response.sendRedirect("print.jsp");//if the main sheet was
selected send to the amin sheet
        }
        if(request.getParameter("sheet").equals("2")){
            response.sendRedirect("print2.jsp");//if the second sheet was
selected redirect to the second sheet
        }
        if(request.getParameter("sheet").equals("3")){
            response.sendRedirect("print3.jsp");//if the third sheet was
selected, redirect to the third sheet
        }
    }
    %>
<div id="navcontainer"><!-- navigation bar -->
    <ul class="navigation">
        <li id="active"><a href="race.jsp">Character Creator</a></li>
        <li><a href="CharacterViewer.jsp" id="current">Character
Viewer</a></li>
    </ul>
</div>
```

```
<h1>Bram's character creator</h1>
<h2>
    Character Viewer
</h2>


---



Bram Williams



Page 404


```

## print.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page import="com.google.appengine.api.datastore.DatastoreServiceFactory"%>
    <%@ page import="com.google.appengine.api.datastore.DatastoreService"%>
    <%@ page import="com.google.appengine.api.datastore.Query"%>
    <%@ page import="com.google.appengine.api.datastore.Entity"%>
    <%@ page import="com.google.appengine.api.datastore.FetchOptions"%>
    <%@ page import="com.google.appengine.api.datastore.Key"%>
    <%@ page import="com.google.appengine.api.datastore.KeyFactory"%>
    <%@ page import="com.google.appengine.api.datastore.Query.Filter"%>
    <%@ page import="com.google.appengine.api.datastore.Query.FilterOperator"%>
    <%@ page import="com.google.appengine.api.datastore.Query.FilterPredicate"%>
    <%@ page
import="com.google.appengine.api.datastore.Query.CompositeFilter"%><!-- imports
datastore classes -->
    <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the page for the first character sheet page-->

<jsp:useBean id="helper" class="helper.Login" scope="page" /><!-- helper bean -->
<html>
<head>
<link rel="stylesheet" type="text/css" href="printDisplay.css">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
<%
DatastoreService datastore =
DatastoreServiceFactory.getDatastoreService(); //datastore
String username = (String)session.getAttribute("username"); //get username for
helper
int save = (Integer)session.getAttribute("save"); //get save slot for helper
coursework player = (coursework)helper.getCharacter(username, save); //get the
player object from the datastore with the helper
    //using the username and save slot as parameters
%>
</head>
<body>

<img id = "sheet" src = "/resources/file-page1.jpg" alt = "character sheet" height
= "1122" width = "794">
<!-- get character sheet image from the resources folder -->

<!-- all the information is printed out in different tags. This is moved to the
correct position on the character sheet with css -->
<!-- this creates a useable character sheet the user can print to play with -->
<h5 id = "name">
<%
try{
    out.println(player.pName);
}catch(Exception e){
}
%>
</h5>
<h5 id = "class">
<%
try{
    out.println(player.Class);

```

```

}catch(Exception e){
}
%>
</h5>
<h5 id = "background">
<%
try{
    out.println(player.background);
}catch(Exception e){
}
%>
</h5>
<h5 id = "pName">
<%
try{
    out.println(player.name);
}catch(Exception e){
}
%>
</h5>
<h5 id = "race">
<%
try{
    out.println(player.race);
}catch(Exception e){
}
%>
</h5>
<h5 id = "alignment">
<%
try{
    out.println(player.alignment);
}catch(Exception e){
}
%>
</h5>
<h5 id = "xp">
<%
try{
    out.println(player.xp);
}catch(Exception e){
}
%>
</h5>
<h5 id = "profBonus">
+2
</h5>
<h5 id = "str">
<%
try{
    out.println(player.Str);
}catch(Exception e){
}
%>
</h5>
<h5 id = "dex">
<%
try{

```

```

        out.println(player.Dex);
}catch(Exception e){
}
%>
</h5>
<h5 id = "con">
<%
try{
    out.println(player.Con);
}catch(Exception e){
}
%>
</h5>
<h5 id = "int">
<%
try{
    out.println(player.Int);
}catch(Exception e){
}
%>
</h5>
<h5 id = "wis">
<%
try{
    out.println(player.Wis);
}catch(Exception e){
}
%>
</h5>
<h5 id = "cha">
<%
try{
    out.println(player.Cha);
}catch(Exception e){
}
%>
</h5>

<!-- if a value is greater than 0 and is a modifier it requires a + sign, the if statements put the +'s in -->

<h5 id = "strm">
<%
try{
    if(player.StrMod > 0){
        out.println("+");
    }
    out.println(player.StrMod);
}catch(Exception e){
}
%>
</h5>
<h5 id = "dexm">
<%
try{
    if(player.DexMod > 0){
        out.println("+");
    }
}

```

```

        out.println(player.DexMod);
}catch(Exception e){
}
%>
</h5>
<h5 id = "conm">
<%
try{
    if(player.ConMod > 0){
        out.println("+");
    }
    out.println(player.ConMod);
}catch(Exception e){
}
%>
</h5>
<h5 id = "intm">
<%
try{
    if(player.IntMod > 0){
        out.println("+");
    }
    out.println(player.IntMod);
}catch(Exception e){
}
%>
</h5>
<h5 id = "wism">
<%
try{
    if(player.WisMod > 0){
        out.println("+");
    }
    out.println(player.WisMod);
}catch(Exception e){
}
%>
</h5>
<h5 id = "cham">
<%
try{
    if(player.ChaMod > 0){
        out.println("+");
    }
    out.println(player.ChaMod);
}catch(Exception e){
}
%>
</h5>
<h5 id = "ac">
<%
try{
    out.println(player.ac);
}catch(Exception e){
}
%>
</h5>
<h5 id = "initiative">

```

```

<%
try{
    if(player.initiative > 0){
        out.println("+");
    }
    out.println(player.initiative);
}catch(Exception e){
}
%>
</h5>
<h5 id = "speed">
<%
try{
    out.println(player.speed);
}catch(Exception e){
}
%>
</h5>
<h4 id = "personality">
<%
try{
    out.println(player.personality);
}catch(Exception e){
}
%>
</h4>
<h4 id = "ideal">
<%
try{
    out.println(player.ideal);
}catch(Exception e){
}
%>
</h4>
<h4 id = "bond">
<%
try{
    out.println(player.bond);
}catch(Exception e){
}
%>
</h4>
<h4 id = "flaw">
<%
try{
    out.println(player.flaw);
}catch(Exception e){
}
%>
</h4>
<h4 id = "features">
<%
try{
    out.println(player.vision + "<br>" + player.features);
}catch(Exception e){
}
%>
</h4>

```

```

<!-- proficiencies require a circle is shaded in. large full stops are used to
shad in the circles -->

<h6 id = "strS">
<%
try{
    if(player.strSave){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "dexS">
<%
try{
    if(player.dexSave){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "conS">
<%
try{
    if(player.conSave){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "wisS">
<%
try{
    if(player.wisSave){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "intS">
<%
try{
    if(player.intSave){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "chaS">
<%
try{
    if(player.chaSave){

```

```

        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h5 id = "strMV">
<%
try{
    if(player.strSaveV > 0){
        out.println("+" + player.strSaveV);
    }else{
        out.println(player.strSaveV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "dexMV">
<%
try{
    if(player.dexSaveV > 0){
        out.println("+" + player.dexSaveV );
    }else{
        out.println(player.dexSaveV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "conMV">
<%
try{
    if(player.conSaveV > 0){
        out.println("+" + player.conSaveV);
    }else{
        out.println(player.conSaveV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "wisMV">
<%
try{
    if(player.wisSaveV > 0){
        out.println("+" + player.wisSaveV);
    }else{
        out.println(player.wisSaveV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "intMV">
<%
try{
    if(player.intSaveV > 0){

```

```

        out.println("+" + player.intSaveV);
    }else{
        out.println(player.intSaveV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "chaMV">
<%
try{
    if(player.chaSaveV > 0){
        out.println("+" + player.chaSaveV);
    }else{
        out.println(player.chaSaveV);
    }
}catch(Exception e){
}
%>
</h5>
<h6 id = "s1">
<%
try{
    if(player.acrobatics){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s2">
<%
try{
    if(player.animalHandling){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s3">
<%
try{
    if(player.arcana){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s4">
<%
try{
    if(player.athletics){
        out.println(".");
    }
}catch(Exception e){
}

```

```

%>
</h6>
<h6 id = "s5">
<%
try{
    if(player.deception){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s6">
<%
try{
    if(player.history){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s7">
<%
try{
    if(player.insight){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s8">
<%
try{
    if(player.intimidation){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s9">
<%
try{
    if(player.investigation){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s10">
<%
try{
    if(player.medicine){
        out.println(".");
    }
}

```

```

}catch(Exception e){
}
%>
</h6>
<h6 id = "s11">
<%
try{
    if(player.nature){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s12">
<%
try{
    if(player.perception){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s13">
<%
try{
    if(player.performance){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s14">
<%
try{
    if(player.persuasion){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s15">
<%
try{
    if(player.religion){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s16">
<%
try{
    if(player.sleightOfHand){

```

```

        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s17">
<%
try{
    if(player.stealth){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h6 id = "s18">
<%
try{
    if(player.survival){
        out.println(".");
    }
}catch(Exception e){
}
%>
</h6>
<h5 id = "v1">
<%
try{
    if(player.acrobaticsV > 0){
        out.println("+" + player.acrobaticsV);
    }else{
        out.println(player.acrobaticsV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v2">
<%
try{
    if(player.animalHandlingV > 0){
        out.println("+" + player.animalHandlingV);
    }else{
        out.println(player.animalHandlingV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v3">
<%
try{
    if(player.arcanaV > 0){
        out.println("+" + player.arcanaV);
    }else{
        out.println(player.arcanaV);
    }
}

```

```

}catch(Exception e){
}
%>
</h5>
<h5 id = "v4">
<%
try{
    if(player.athleticsV > 0){
        out.println("+" + player.athleticsV);
    }else{
        out.println(player.athleticsV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v5">
<%
try{
    if(player.deceptionV > 0){
        out.println("+" + player.deceptionV);
    }else{
        out.println(player.deceptionV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v6">
<%
try{
    if(player.historyV > 0){
        out.println("+" + player.historyV);
    }else{
        out.println(player.historyV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v7">
<%
try{
    if(player.insightV > 0){
        out.println("+" + player.insightV);
    }else{
        out.println(player.insightV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v8">
<%
try{
    if(player.intimidationV > 0){
        out.println("+" + player.intimidationV);
    }else{

```

```

        out.println(player.intimidationV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v9">
<%
try{
    if(player.investigationV > 0){
        out.println("+" + player.investigationV);
    }else{
        out.println(player.investigationV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v10">
<%
try{
    if(player.medicineV > 0){
        out.println("+" + player.medicineV);
    }else{
        out.println(player.medicineV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v11">
<%
try{
    if(player.natureV > 0){
        out.println("+" + player.natureV);
    }else{
        out.println(player.natureV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v12">
<%
try{
    if(player.perceptionV > 0){
        out.println("+" + player.perceptionV);
    }else{
        out.println(player.perceptionV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v13">
<%
try{
    if(player.performanceV > 0){

```

```

        out.println("+" + player.performanceV);
    }else{
        out.println(player.performanceV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v14">
<%
try{
    if(player.persuasionV > 0){
        out.println("+" + player.persuasionV);
    }else{
        out.println(player.persuasionV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v15">
<%
try{
    if(player.religionV > 0){
        out.println("+" + player.religionV);
    }else{
        out.println(player.religionV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v16">
<%
try{
    if(player.sleightOfHandV > 0){
        out.println("+" + player.sleightOfHandV);
    }else{
        out.println(player.sleightOfHandV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v17">
<%
try{
    if(player.stealthV > 0){
        out.println("+" + player.stealthV);
    }else{
        out.println(player.stealthV);
    }
}catch(Exception e){
}
%>
</h5>
<h5 id = "v18">
<%

```

```

try{
    if(player.survivalV > 0){
        out.println("+" + player.survivalV);
    }else{
        out.println(player.survivalV);
    }
}catch(Exception e){
}
%>
</h5>

<h5 id = "passiveP">
<%
try{
    out.println(player.passivePerception);
}catch(Exception e){
}
%>
</h5>
<h4 id = "other">
<%
try{
    out.println(player.otherProficiencies + "<br>" + player.languages);
}catch(Exception e){
}
%>
</h4>
<h4 id = "equipment">
<%
try{
    out.println(player.equipment);
}catch(Exception e){
}
%>
</h4>
<h5 id = "hp">
<%
try{
    out.println(player.hitPoints);
}catch(Exception e){
}
%>
</h5>
<h5 id = "hd">
<%
try{
    out.println(player.hitDice);
}catch(Exception e){
}
%>
</h5>
<h5 id = "money">
<%
try{
    out.println(player.money);
}catch(Exception e){
}
%>

```

```
</h5>
<a href="javascript:window.print()" id = "btn">click this link, right click and
print, or use ctrl + p</a><!-- this button prints the page -->
</body>
</html>
```

## print2.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page import="com.google.appengine.api.datastore.DatastoreServiceFactory"%>
    <%@ page import="com.google.appengine.api.datastore.DatastoreService"%>
    <%@ page import="com.google.appengine.api.datastore.Query"%>
    <%@ page import="com.google.appengine.api.datastore.Entity"%>
    <%@ page import="com.google.appengine.api.datastore.FetchOptions"%>
    <%@ page import="com.google.appengine.api.datastore.Key"%>
    <%@ page import="com.google.appengine.api.datastore.KeyFactory"%>
    <%@ page import="com.google.appengine.api.datastore.Query.Filter"%>
    <%@ page import="com.google.appengine.api.datastore.Query.FilterOperator"%>
    <%@ page import="com.google.appengine.api.datastore.Query.FilterPredicate"%>
    <%@ page
import="com.google.appengine.api.datastore.Query.CompositeFilter"%><!-- imports
datastore classes -->
    <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the page for the second character sheet page-->

<jsp:useBean id="helper" class="helper.Login" scope="page" /><!-- helper bean -->
<html>
<head>
<link rel="stylesheet" type="text/css" href="printDisplay.css">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
<%
DatastoreService datastore = DatastoreServiceFactory.getDatastoreService();
String username = (String)session.getAttribute("username");
int save = (Integer)session.getAttribute("save");
coursework player = (coursework)helper.getCharacter(username, save); //gets the
correct character as an object
%>
</head>
<body>

<img id = "sheet" src = "/resources/file-page2.jpg" alt = "character sheet" height
= "1122" width = "794">
<!-- displays the second character sheet from the resources folder -->

<!-- tags display the information for this page using css to put them in the right
location -->
<h5 id = "name2">
<%
try{
    out.println(player.pName);
}catch(Exception e){
}
%>
</h5>
<h5 id = "age">
<%
try{
    out.println(player.age);
}catch(Exception e){
}
%>
</h5>
```

```

<h5 id = "height">
<%
try{
    out.println(player.realH);
}catch(Exception e){
}
%>
</h5>
<h5 id = "weight">
<%
try{
    out.println(player.realW);
}catch(Exception e){
}
%>
</h5>
<h5 id = "eyes">
<%
try{
    out.println(player.eyes);
}catch(Exception e){
}
%>
</h5>
<h5 id = "skin">
<%
try{
    out.println(player.skin);
}catch(Exception e){
}
%>
</h5>
<h5 id = "hair">
<%
try{
    out.println(player.hair);
}catch(Exception e){
}
%>
</h5>
<h4 id = "backstory">
<%
try{
    out.println(player.backstory);
}catch(Exception e){
}
%>
</h4>
<a href="javascript:window.print()" id = "btn">click this link, right click and
print, or use ctrl + p</a><!-- this button prints the page -->
</body>
</html>

```

### print3.jsp

```
<%@page import="cc.coursework, cc.processing"%>
<%@ page import="com.google.appengine.api.datastore.DatastoreServiceFactory"%>
    <%@ page import="com.google.appengine.api.datastore.DatastoreService"%>
    <%@ page import="com.google.appengine.api.datastore.Query"%>
    <%@ page import="com.google.appengine.api.datastore.Entity"%>
    <%@ page import="com.google.appengine.api.datastore.FetchOptions"%>
    <%@ page import="com.google.appengine.api.datastore.Key"%>
    <%@ page import="com.google.appengine.api.datastore.KeyFactory"%>
    <%@ page import="com.google.appengine.api.datastore.Query.Filter"%>
    <%@ page import="com.google.appengine.api.datastore.Query.FilterOperator"%>
    <%@ page import="com.google.appengine.api.datastore.Query.FilterPredicate"%>
    <%@ page
import="com.google.appengine.api.datastore.Query.CompositeFilter"%><!-- import
datastore classes -->
    <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!--the page for the spellcasting character sheet page-->

<jsp:useBean id="helper" class="helper.Login" scope="page" />
<html>
<head>
<link rel="stylesheet" type="text/css" href="printDisplay.css">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>character creator</title>
<%
DatastoreService datastore = DatastoreServiceFactory.getDatastoreService();
String username = (String)session.getAttribute("username");
int save = (Integer)session.getAttribute("save");
coursework player = (coursework)helper.getCharacter(username, save); //get correct
save as an object
if(player.spellDC == 111){ //default spellDC if the class is not a spellcaster
    response.sendRedirect("CharacterViewer.jsp"); //if the player is not a
spellcaster redirect away from the page
}
%>
</head>
<body>

<img id = "sheet" src = "/resources/file-page3.jpg" alt = "character sheet" height
= "1122" width = "794">
<!-- get character spellcasting sheet image from the resources folder -->

<!-- output tags to the correct place on the page with css -->
<h5 id = "Sclass">
<%
try{
    out.println(player.Class);
}catch(Exception e){
}
%>
</h5>
<h5 id = "Sabil">
<%
try{
    out.println(player.spellAbil);
}catch(Exception e){
```

```

}
%>
</h5>
<h5 id = "SDC">
<%
try{
    out.println(player.spellDC);
}catch(Exception e){
}
%>
</h5>
<h5 id = "Sbonus">
<%
try{
    if(player.spellMod>0){
        out.println("+" +player.spellMod);
    }else{
        out.println(player.spellMod);
    }
}catch(Exception e){
}
%>
</h5>
<a href="javascript:window.print()" id = "btn">click this link, right click and
print, or use ctrl + p</a><!-- this button prints the page -->
</body>
</html>

```

```

courseworkApp.css
/*css for all the main pages, themed to keep pages centralised with headings in
pastel purple*/

body{
    font-family: sans-serif; /*font for all the html*/
}
#Log{
    margin-right: 45%;
    text-align: right;
}
#characterName{
    position: relative;
    left: -20px;
    top: -500px;
}
#save{
    text-align: center;
}
#feet{
width: 30px;
}
#inches{
width: 30px;
}
#Weight{
width: 30px;
}

ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
}

li {
    float: left; /*puts the navigation bar at the top left*/
}

a {
    display: block;
    width: 150px;
}
a.align{
    text-align: center;
    margin: auto;
}
#page1{
    float: left;
    margin-left: 2%;
    width: 48%;
}
#page2{
    float: left;
    margin-right: 2%;
    width: 48%;
}

```

```
#leftcolumn {
    margin-left: 30%;
    width: 18%;
    float: left;
    text-align: center;
}
#age{
    position: relative;
    right: 18px
}
#eyes{
    position: relative;
    right: 21px
}
#skin{
    position: relative;
    right: 19px
}
#hair{
    position: relative;
    right: 18px
}
#rightcolumn {
    margin-right: 30%;
    width: 18%;
    float: left;
    text-align: center;
}
#control{
    width: 700px;
    height: 100px;

    position: absolute;
    top:0;
    bottom: 0;
    left: 0;
    right: 0;

    margin: auto;
}
#leftcolumn1 {

    position: fixed;
    top:25%;
    left: 21%;
    text-align: center;
}
#submit{
    position: fixed;
    top:50%;
    right: 47%;
    text-align: center;
}
#rightcolumn1 {

    position: fixed;
    top:25%;
    right: 31%;
```

```

        text-align: center;
    }
#fixedbutton {
}
/*hidden radio buttons for passing js values to jsp*/
#p1{
    visibility: hidden;
}
#p2{
    visibility: hidden;
}
#p3{
    visibility: hidden;
}
#p4{
    visibility: hidden;
}
#p5{
    visibility: hidden;
}
#p6{
    visibility: hidden;
}
#p7{
    visibility: hidden;
}
#align{
    text-align: center;
}
b{
    color: #8F568F; /*purple*/
}
p{
    color: #8F568F;
    font-weight: bold; /*puts all p tags in bold*/
}
h1 {
    font-size: 2em;
    font-weight: bold;
    color: #8F568F;
    margin: 40px 0px 2%;
    text-align: center;
}
h2 {
    font-size: 2em;
    font-weight: bold;
    color: #8F568F;
    margin: 40px 0px 2%;
    text-align: center;
}
h3 {
    color: #8F568F;
    text-align: center;
}
.sendButton {
    display: block;

```

```
    font-size: 16pt;
}

/** Most GWT widgets already have a style name defined */
.gwt-DialogBox {
    width: 400px;
}

.dialogVPanel {
    margin: 5px;
}

.serverResponseLabelError {
    color: red;
}

/** Set ids using widget.getElement().setId("idOfElement") */
#closeButton {
    margin: 15px 6px 6px;
}
```

```
printDisplay.css
@CHARSET "ISO-8859-1";
/*absolute positioning to put all the id's in the correct place on their
respective print pages*/
#name{
position: absolute;
z-index: 2;
top:70px;
left:110px;
}
#name2{
position: absolute;
z-index: 2;
top:75px;
left:110px;
}
#class{
position: absolute;
z-index: 2;
top:50px;
left:370px;
}
#age{
position: absolute;
z-index: 2;
top:55px;
left:360px;
}
#background{
position: absolute;
z-index: 2;
top:50px;
left:510px;
}
#height{
position: absolute;
z-index: 2;
top:55px;
left:500px;
}
#pName{
position: absolute;
z-index: 2;
top:50px;
left:635px;
}
#weight{
position: absolute;
z-index: 2;
top:55px;
left:630px;
}
#race{
position: absolute;
z-index: 2;
top:88px;
left:370px;
}
```

```
#eyes{
position: absolute;
z-index: 2;
top:93px;
left:360px;
}
#alignment{
position: absolute;
z-index: 2;
top:88px;
left:510px;
}
#skin{
position: absolute;
z-index: 2;
top:93px;
left:500px;
}
#xp{
position: absolute;
z-index: 2;
top:88px;
left:640px;
}
#hair{
position: absolute;
z-index: 2;
top:93px;
left:630px;
}
#profBonus{
    position: absolute;
    z-index: 2;
    top:222px;
    left:142px;
}
#strm{
    position: absolute;
    z-index: 2;
    top:210px;
    left:74px;
}
#dexm{
    position: absolute;
    z-index: 2;
    top:311px;
    left:74px;
}
#conm{
    position: absolute;
    z-index: 2;
    top:412px;
    left:74px;
}
#intm{
    position: absolute;
    z-index: 2;
    top:513px;
```

```
        left:74px;
    }
#wism{
    position: absolute;
    z-index: 2;
    top:614px;
    left:74px;
}
#cham{
    position: absolute;
    z-index: 2;
    top:715px;
    left:74px;
}
#str{
    position: absolute;
    z-index: 2;
    top:245px;
    left:75px;
}
#dex{
    position: absolute;
    z-index: 2;
    top:346px;
    left:75px;
}
#con{
    position: absolute;
    z-index: 2;
    top:447px;
    left:75px;
}
#int{
    position: absolute;
    z-index: 2;
    top:549px;
    left:75px;
}
#wis{
    position: absolute;
    z-index: 2;
    top:651px;
    left:75px;
}
#cha{
    position: absolute;
    z-index: 2;
    top:751px;
    left:75px;
}
#ac{
    position: absolute;
    z-index: 2;
    top:192px;
    left:323px;
}
#initiative{
    position: absolute;
```

```
    z-index: 2;
    top:195px;
    left:394px;
}
#speed{
    position: absolute;
    z-index: 2;
    top:195px;
    left:472px;
}
#personality{
    position: absolute;
    z-index: 2;
    top:185px;
    left:556px;
    width: 180px;
    line-height: 18px;
}
#ideal{
    position: absolute;
    z-index: 2;
    top:282px;
    left:556px;
    width: 180px;
    line-height: 17px;
}
#bond{
    position: absolute;
    z-index: 2;
    top:360px;
    left:556px;
    width: 180px;
    line-height: 17px;
}
#flaw{
    position: absolute;
    z-index: 2;
    top:438px;
    left:556px;
    width: 180px;
    line-height: 17px;
}
#passiveP{
    position: absolute;
    z-index: 2;
    top:820px;
    left:56px;
    width: 220px;
    line-height: 16px;
}
#features{
    position: absolute;
    z-index: 2;
    top:531px;
    left:543px;
    width: 220px;
    line-height: 15.5px;
}
```

```
#other{
    position: absolute;
    z-index: 2;
    top:874px;
    left:52px;
    width: 220px;
    line-height: 15.5px;
}
#equipment{
    position: absolute;
    z-index: 2;
    top:828px;
    left:357px;
    width: 156px;
    line-height: 15.5px;
}
#strs{
    position: absolute;
    z-index: 2;
    top:147.2px;
    left:137.5px;
}
#dexS{
    position: absolute;
    z-index: 2;
    top:166px;
    left:137.5px;
}
#cons{
    position: absolute;
    z-index: 2;
    top:185.2px;
    left:137.5px;
}
#ints{
    position: absolute;
    z-index: 2;
    top:204.3px;
    left:137.5px;
}
#wiss{
    position: absolute;
    z-index: 2;
    top:223.3px;
    left:137.5px;
}
#chas{
    position: absolute;
    z-index: 2;
    top:242.5px;
    left:137.5px;
}
#s1{
    position: absolute;
    z-index: 2;
    top:310.4px;
    left:137.5px;
```

```
}

#s2{
    position: absolute;
    z-index: 2;
    top:329.3px;
    left:137.5px;
}

#s3{
    position: absolute;
    z-index: 2;
    top:348.5px;
    left:137.5px;
}

#s4{
    position: absolute;
    z-index: 2;
    top:367.5px;
    left:137.5px;
}

#s5{
    position: absolute;
    z-index: 2;
    top:386.5px;
    left:137.5px;
}

#s6{
    position: absolute;
    z-index: 2;
    top:406px;
    left:137.5px;
}

#s7{
    position: absolute;
    z-index: 2;
    top:425px;
    left:137.5px;
}

#s8{
    position: absolute;
    z-index: 2;
    top:444px;
    left:137.5px;
}

#s9{
    position: absolute;
    z-index: 2;
    top:463.2px;
    left:137.5px;
}

#s10{
    position: absolute;
    z-index: 2;
    top:482.2px;
    left:137.5px;
}

#s11{
    position: absolute;
    z-index: 2;
```

```
    top:501.4px;
    left:137.5px;
}
#s12{
    position: absolute;
    z-index: 2;
    top:520.6px;
    left:137.5px;
}
#s13{
    position: absolute;
    z-index: 2;
    top:539.8px;
    left:137.5px;
}
#s14{
    position: absolute;
    z-index: 2;
    top:558.8px;
    left:137.5px;
}
#s15{
    position: absolute;
    z-index: 2;
    top:578px;
    left:137.5px;
}
#s16{
    position: absolute;
    z-index: 2;
    top:597.2px;
    left:137.5px;
}
#s17{
    position: absolute;
    z-index: 2;
    top:616.2px;
    left:137.5px;
}
#s18{
    position: absolute;
    z-index: 2;
    top:635.3px;
    left:137.5px;
}
#v1{
    position: absolute;
    z-index: 2;
    top:429.6px;
    left:157px;
}
#v2{
    position: absolute;
    z-index: 2;
    top:448.7px;
    left:157px;
}
#v3{
```

```
    position: absolute;
    z-index: 2;
    top:467.8px;
    left:157px;
}
#v4{
    position: absolute;
    z-index: 2;
    top:486.8px;
    left:157px;
}
#v5{
    position: absolute;
    z-index: 2;
    top:506.3px;
    left:157px;
}
#v6{
    position: absolute;
    z-index: 2;
    top:525.3px;
    left:157px;
}
#v7{
    position: absolute;
    z-index: 2;
    top:544.3px;
    left:157px;
}
#v8{
    position: absolute;
    z-index: 2;
    top:563.5px;
    left:157px;
}
#v9{
    position: absolute;
    z-index: 2;
    top:582.5px;
    left:157px;
}
#v10{
    position: absolute;
    z-index: 2;
    top:601.7px;
    left:157px;
}
#v11{
    position: absolute;
    z-index: 2;
    top:620.9px;
    left:157px;
}
#v12{
    position: absolute;
    z-index: 2;
    top:640.2px;
    left:157px;
```

```
}

#v13{
    position: absolute;
    z-index: 2;
    top:659.1px;
    left:157px;
}

#v14{
    position: absolute;
    z-index: 2;
    top:678.3px;
    left:157px;
}

#v15{
    position: absolute;
    z-index: 2;
    top:697.5px;
    left:157px;
}

#v16{
    position: absolute;
    z-index: 2;
    top:716.6px;
    left:157px;
}

#v17{
    position: absolute;
    z-index: 2;
    top:735.6px;
    left:157px;
}

#v18{
    position: absolute;
    z-index: 2;
    top:754.7px;
    left:157px;
}

#strMV{
    position: absolute;
    z-index: 2;
    top:268px;
    left:157px;
}

#dexMV{
    position: absolute;
    z-index: 2;
    top:287.2px;
    left:157px;
}

#conMV{
    position: absolute;
    z-index: 2;
    top:306.4px;
    left:157px;
}

#intMV{
    position: absolute;
    z-index: 2;
```

```

        top:325.6px;
        left:157px;
    }
#wisMV{
    position: absolute;
    z-index: 2;
    top:344.8px;
    left:157px;
}
#chaMV{
    position: absolute;
    z-index: 2;
    top:364px;
    left:157px;
}
h6{
    font-size: 300%;
    font-family: Times New Roman;
}
#hp{
    position: absolute;
    z-index: 2;
    top:257px;
    left:393px;
}
#hd{
    position: absolute;
    z-index: 2;
    top:428px;
    left:332px;
}
#money{
    position: absolute;
    z-index: 2;
    top:941px;
    left:316px;
}
#backstory{
    position: absolute;
    z-index: 2;
    top:528px;
    left:63px;
    width: 200px;
    line-height: 16px;
}
#ScClass{
    position: absolute;
    z-index: 2;
    top:75px;
    left:100px;
}
#Sabil{
    position: absolute;
    z-index: 2;
    top:68px;
    left:380px;
}

```

```

#SDC{
    position: absolute;
    z-index: 2;
    top:68px;
    left:540px;
}
#Sbonus{
    position: absolute;
    z-index: 2;
    top:68px;
    left:675px;
}
#sheet{
position: absolute;
top: 0;
z-index: -1;
}
h5{
    /*
    color: #CCCCCC
    */
}
h4{
    font-size: 60%;
}
#btn{
    position: absolute;
    top: 1110px;
    z-index: 1;
}
body{
    font-family: sans-serif;
}
@media print {
    #Header, #Footer { display: none !important; }
    #btn { display: none !important; } /*on print hides unwanted parts of the page*/
}

```

## Appengine-web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">

  <application>whsbrpg</application>
  <version>1</version>

  <!--
    Allows App Engine to send multiple requests to one instance in parallel:
  -->
  <threadsafe>true</threadsafe>

  <!-- Configure serving/caching of GWT files -->
  <static-files>
    <include path="**" />

    <!-- The following line requires App Engine 1.3.2 SDK -->
    <include path="*.nocache.*" expiration="0s" />

    <include path="*.cache.*" expiration="365d" />
    <exclude path="*.gwt.rpc" />
  </static-files>

  <!-- Configure java.util.logging -->
  <system-properties>
    <property name="java.util.logging.config.file" value="WEB-
INF/logging.properties"/>
  </system-properties>
  <sessions-enabled>true</sessions-enabled>
  <!--

    It's possible to reduce request latency by configuring your application to
    asynchronously write HTTP session data to the datastore:

    <async-session-persistence enabled="true" />

    With this feature enabled, there is a very small chance your app will see
    stale session data. For details, see

    http://code.google.com/appengine/docs/java/config/appconfig.html#Enabling\_Sessions
    -->

  </appengine-web-app>
```

## Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
        http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    version="2.5"
    xmlns="http://java.sun.com/xml/ns/javaee">

    <!-- Servlets -->
    <servlet>
        <servlet-name>greetServlet</servlet-name>
        <servlet-class>bramrethCC.server.GreetingServiceImpl</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>greetServlet</servlet-name>
        <url-pattern>/appspot/greet</url-pattern>
    </servlet-mapping>

    <!-- Default page to serve -->
    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>

</web-app>
```

## **URL**

<http://whsbrpg.appspot.com/>