Jonathan Barnett

As part of the Technical Progression Certificate, students will complete a project which has been written by Weston College in collaboration with an employer partner to ensure industry relevance. This project asks students to develop a D&D Character Creation program.

Employer Set Project 2021/22

L2 Technical Progression Certificate in Computing and Digital (Transition Programme)

Contents

[Activities 9](#_Toc100820289)

[Task 1 - Flowchart 9](#_Toc100820290)

[Task 2 - Pseudocode 9](#_Toc100820291)

[Task 3 – Test Plan 9](#_Toc100820292)

[Task 4 – Code & Test 9](#_Toc100820293)

[Task 5 - Evaluate 9](#_Toc100820294)

[Appendix 1 – Testing Template 10](#_Toc100820295)

Scenario

You are a junior software developer and have been tasked by the company you work for to design, develop, test and document the following program requested by the client

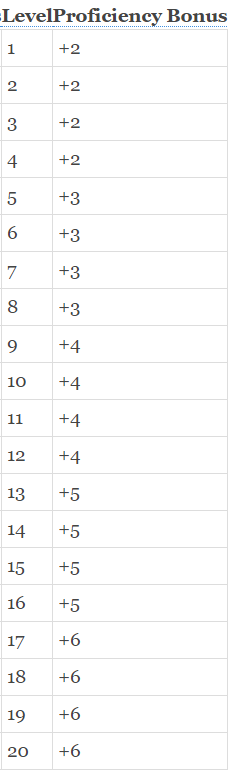
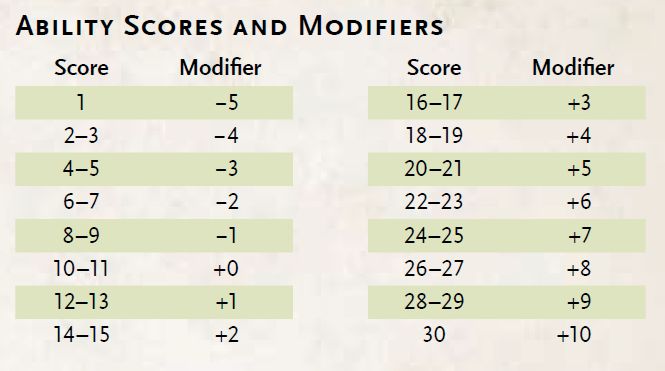
The program is designed to allow for handling of character and party attributes in a game. This should be developed in an Object Oriented method.

This program needs to have a menu system which allows for access to the following:

* Character Creation
* Party Creation
* GM Viewer

Firstly the creation of a character, this should provide the user with prompts where they input details about the character they are making. This must include:

(If a statement does not include an explanation assume an appropriate field data type)

* Character Name
* Player Name
* Race
  + Dragonborn
    - +2 Strength, +1 Charisma
  + Dwarf
    - +2 Constitution, Darkvision Trait
  + Elf
    - +2 Dexterity, Darkvision Trait
  + Gnome
    - +2 Intelligence, Darkvision Trait
  + Half-Elf
    - +2 Charisma, +1 to two other ability score, Darkvision Trait
  + Halfling
    - +2 Dexterity
  + Half-Orc
    - +2 Strength, +1 Constitution, Darkvision Trait
  + Human
    - +1 to ALL abilities
  + Tiefling
    - +2 Charisma, +1 Intelligence, Darkvision Trait
* Class
* A player need to choose their class as well as the skills they wish to be good at within that class
  + Barbarian (d12)
    - Saving Throws: Strength, Constitution
    - Choose two from Animal Handling, Athletics, Intimidation, Nature, Perception, and Survival
  + Bard (d8)
    - Saving Throws: Dexterity, Charisma
    - Skills: Choose any three
  + Cleric (d8)
    - Saving Throws: Wisdom, Charisma
    - Skills: Choose two from History, Insight, Medicine, Persuasion, and Religion
  + Druid (d8)
    - Saving Throws: Intelligence, Wisdom
    - Skills: Choose two from Arcana, Animal Handling, Insight, Medicine, Nature, Perception, Religion, and Survival
  + Fighter (d10)
    - Saving Throws: Strength, Constitution
    - Skills: Choose two skills from Acrobatics, Animal Handling, Athletics, History, Insight, Intimidation, Perception, and Survival
  + Monk (d8)
    - Saving Throws: Strength, Dexterity
    - Skills: Choose two from Acrobatics, Athletics, History, Insight, Religion, and Stealth
  + Paladin (d10)
    - Saving Throws: Wisdom, Charisma
    - Skills: Choose two from Athletics, Insight, Intimidation, Medicine, Persuasion, and Religion
  + Ranger (d10)
    - Saving Throws: Strength, Dexterity
    - Skills: Choose three from Animal Handling, Athletics, Insight, Investigation, Nature, Perception, Stealth, and Survival
  + Rogue (d8)
    - Saving Throws: Dexterity, Intelligence
    - Skills: Choose four from Acrobatics, Athletics, Deception, Insight, Intimidation, Investigation, Perception, Performance, Persuasion, Sleight of Hand, and Stealth
  + Sorcerer (d6)
    - Saving Throws: Constitution, Charisma
    - Skills: Choose two from Arcana, Deception, Insight, Intimidation, Persuasion, and Religion
  + Warlock (d8)
    - Saving Throws: Wisdom, Charisma
    - Skills: Choose two skills from Arcana, Deception, History, Intimidation, Investigation, Nature, and Religion
  + Wizard (d6)
    - Saving Throws: Intelligence, Wisdom
    - Skills: Choose two from Arcana, History, Insight, Investigation, Medicine, and Religion
* Level
  + This is set by the user
  + This cannot be a number larger than 20
* Proficiency
  + This improves with level in line with the following table
  + 
* HP
  + This is determined by level, at level 1 they have the maximum amount of HP for their dice i.e a Monk @ level 1 would have 8hp. Each level above this is rolled for, so for the Monk each level above 1 would be a virtual dice roll on a d8. The HP is changed based on the Constitution stat, in line with the ability score improvement.
* Ability Score
  + Strength
  + Dexterity
  + Constitution
  + Wisdom
  + Intelligence
  + Charisma
  + This is determined by a random roll. The program should “roll” 4 d6 dice, and display the numbers. It should then total the 3 highest from each set of rolls. These are added into a pool that the player can then choose how to allocate. For example, if the computer generated the roll 5,5,3,1 then the stat value would be 13, this would be saved into a pool that the player can then choose where to allocate
* 
* Skills
  + These have a governing attribute which uses the modifier table to determine the amount that they go up by + their proficiency. For example a character with 19 Strength and +3 proficiency would have a 7 in athletics, if they are proficient in it.
  + Strength
    - Athletics
  + Dexterity
    - Acrobatics
    - Sleight of Hand
    - Stealth
  + Intelligence
    - Arcana
    - History
    - Investigation
    - Nature
    - Religion
  + Wisdom
    - Animal Handling
    - Insight
    - Medicine
    - Perception
    - Survival
  + Charisma
    - Deception
    - Intimidation
    - Performance
    - Persuasion
* Saving throws
  + Every class can make a saving throw for each attribute which is determined by their raw ability score modifier, UNLESS they are proficient in which case they also add their proficiency bonus
* Passive perception and insight
  + These are determined from the characters perception and insight score +10
* Character Armour Class Number
* Character Background
* The number of Hit Dice
  + This is determined by level on a 1:1 ratio

Secondly the program should be able to build a party made up of the characters that have been created. This is to allow the Game Master to track different pieces of information.

* A set of characters (min of 1) must exist to access this menu
* A party name needs to be set
* Characters that have been made are then added to the party
* The same character cannot be added twice

Third the GM view, this allows the GM to look up different stats about the party, once these have been viewed it should then clear the screen and return to the menu.

The requirements set out that the following needs to be viewed

* List of party character names and player names
* Display individual party armour class
* Display individual party saving throws for a chosen save
* Display skill modifiers for each member for a chosen skill
* Display current party passive insight and perception, These should be displayed in order from highest to lowest
* Display individual party HP
* Display which members in the party have the darkvision trait and which ones do not

This project needs to include full documentation detailing the design process, this should include a flowchart of the elements and pseudocode planning. Additionally, evidence of thorough testing needs to be shown throughout.

You must design, implement and test your program.

You must also justify and evaluate your decisions.

When you are designing and developing the solution ensure that:

• standard programming conventions have been followed

• it is efficient and robust

• it is user friendly

Any questions regarding the development of the project should be fielded through Teams to the client via Jon Barnett

Activities

Task 1 - Flowchart

Produce a flow chart, using British Computing Society symbols, to plan the logic and

processes for the program.

Task 2 - Pseudocode

Produce pseudocode that a software developer could use to create the program.

Task 3 – Test Plan

Use the table in appendix 1 to plan the testing of your complete program.

You should complete these columns:

* Test number
* Purpose of test
* Test data
* Expected result.

You will need to save this file so that you can complete the testing in activity 4.

Task 4 – Code & Test

Use your flow chart, pseudocode and test plan to help write and test your program.

You should:

* write a program that meets the scenario requirements
* test your solution throughout the development process to ensure that it functions
* as expected
* record the outcomes of your testing and any actions taken in your test plan
* document.

Your evidence should include:

* a copy of your code containing annotations/comments
* a copy of your fully completed test plan document.

Task 5 - Evaluate

Evaluate your program solution.

You should cover:

* how well your solution meets the requirements of the scenario
* the quality and performance of your program
* the choices you made about coding conventions
* the changes you made during the development process.

# Appendix 1 – Testing Template

Use this table when you are completing Task 3 & Task 4. You will need to add more rows to the table to allow you to plan and test the whole of your program.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test number | Purpose of test | Test Data | Expected Outcome | Actual Outcome | Comments |
|  |  |  |  |  |  |
|  |  |  |  |  |  |