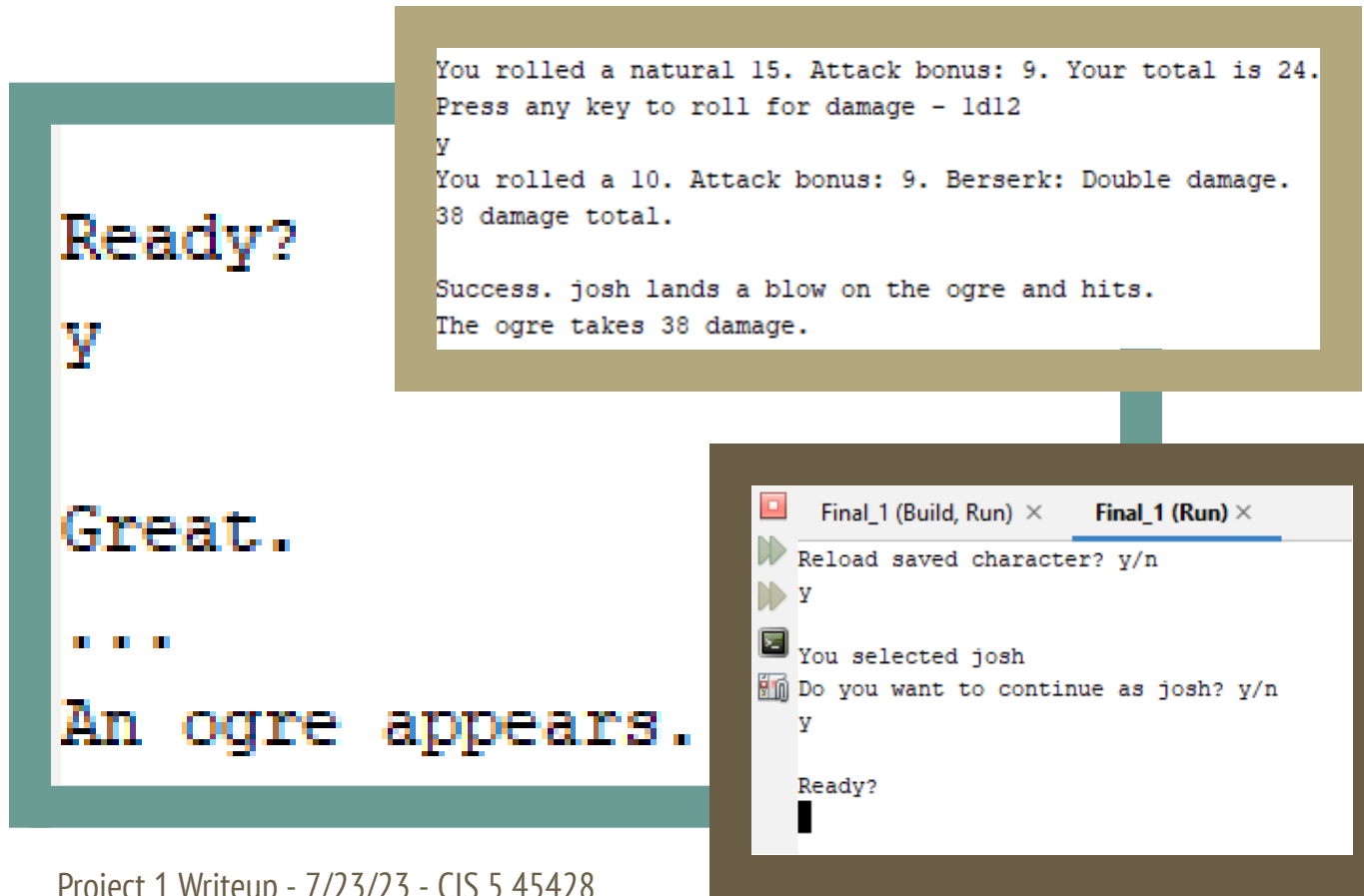


Simplified D&D



Project 1 Writeup - 7/23/23 - CIS 5 45428

Kaylin Nguyen - Simplified DnD - 500+ lines (~650)

Overview

A text based game of a simplified version of Dungeons and Dragons (DnD), meaning it only contains a simple DnD turn based battle mechanic using dice rolls, as well as limited rules, options, and mechanisms. DnD is a tabletop RPG game that utilizes dice rolls to evaluate the success level of a player's action.

Gameplay

You start by creating your character, by choosing their class and allocating their stats, or by loading one in if you have already created one. When you are done, you enter a battle with an enemy, choosing your actions, which can include attacks or powerups, and rolling for the success of your action, as well as rolling for any damage you output, in a turn based format, until either you or your enemy falls to or below 0 hp.

```
Output - Final_1 (Run) x
Reload saved character? y/n
n
Creating character.
Enter character name.
Guy Guy
Your character's name is Guy Guy. Is this correct? y/n
y
Choose your class.
```

```
You selected wizard.
Is this correct? y/n
y
```

```
HP: 30 AC: 12 Attack bonus: 3
con: 0 dex: 2 str: 0 int: 3
Points remaining: 5
Select stat to assign points.
```

```
You selected fighter.
Is this correct? y/n
y
```

```
HP: 60 AC: 16 Attack bonus: 4
con: 3 dex: 0 str: 2 int: 0
Points remaining: 5
Select stat to assign points.
```

```
Input number of points from 0 to 5
5
```

```
5 points. Is this correct? y/n
y
```

```
HP: 30 AC: 12 Attack bonus: 8
```

In character creation, the class you choose affects the actions you can take in combat as well as your base stats, and the stats you allocate can further affect your health points (hp), armor class, and attack bonuses which are added to your rolls.

```
Output - Final_1 (Run) x
y
Enemy HP: [|||||] 100.00 %
Player HP: 55/55
Turn: 1
The ogre reaches out.
Critical failure. The ogre trips and hits itself, losing 12 HP.
Your turn. Press any key to continue.
```

In combat, attack roll totals, including any stat based bonuses, must beat the armor class to hit. If the attack roll is a 20, the hit classifies as a critical hit, and doubles the damage. If the attack roll is 1, it classifies as a critical failure, and the damage is redirected towards the attacker. In either case, you roll for the damage done, and that damage is subtracted from the total hp of the recipient.

Powerups only last for a certain number of rounds, with a counter that decreases each round, and provide a bonus to the player while it is active.

```
Mage Armor active. You have 20 AC for 1 rounds after this one remaining.
Choose an action.
1: fireball 3d6 + 3
```

```
Berserk active. You have double damage for 2 rounds after this one remaining.
Choose an action.
1: greatsword 1d12 + 9
```

The enemy attacks first in the round, and then the player takes their turn, until either the enemy or the player falls to or below 0 hp.

```
Enemy HP: [|||||_] 60.00 %
Player HP: 45/55
Turn: 7
```

```
Enemy HP: -4. Success!
The ogre has been slain.
Your name, Guy Guy, will be memorialized in ballads far and wide.
RUN SUCCESSFUL (total time: 6m 15s)
```

Versions

V1

Planning program. Testing random numbers and dice rolling, testing bounds.

V2

Pseudocode and logic for character creation, focus on do while validation.

V3

Pseudocode and logic for actions and dice rolls, focus on switching choices and if.

V4

Testing file i/o and adding to character creation logic. Completing pseudocode.

V5

Writing code based on pseudocode for character creation. Testing character creation module, debugging negative calculation error. Implementing file i/o.

V6

Testing file i/o, debugging string input with spaces. Writing code for fight mechanism. Developing end screen death toggle, updating health bar, and enemy attacking.

V7

Finishing writing code for fight mechanism. Player character actions for fighter and wizard. Debugging missing calculations and char input. Adjusting player experience, completing code.

Final Version - V8

Polishing code. Minor adjustments to player experience. Finished product.

Future Development Notes

A necessary improvement would be to include functions to streamline the code, such as a function to roll dice. Aside from this, arrays could possibly aid in organizing weapons, outputting richer dialogue options, or for expanding the game and allowing the player to choose from various enemies. Additionally, sorting could help enhance and organize the character creation screen.

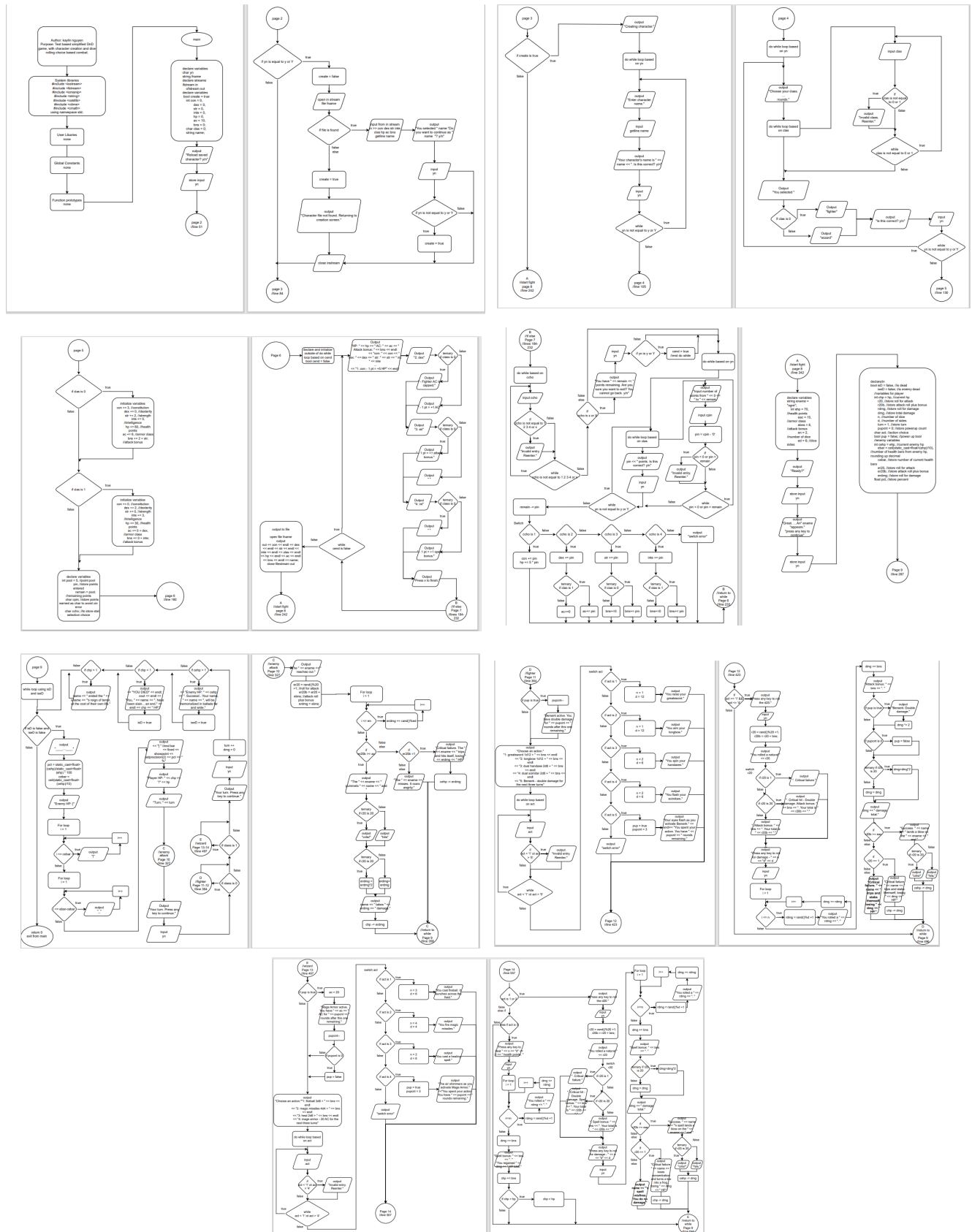
Checkoff Sheet

Cross Reference for Project 1

You are to fill-in with where located in code

Chapter	Section	Topic	Where Line #'s	Pts	Notes
2	2	cout	48, 78, 88, 98, 107, 117, 1	1	prompts, menus
	3	libraries	8, 9, 28-29, 56, 58, 6	8	all used: iostream, iomanip, cmath, csdlib, tstream, string, ctime
	4	variables/literals	36-42, 153-155, 245-249,		No variables in global area, failed project!
	5	identifiers	36-42, 153-155, 245-249,	^^	
	6	Integers	36-42, 153-155, 245	3	
	7	Characters	26, 43, 156, 157, 27	3	
	8	Strings	31, 44, 244	3	
	9	Floats No Doubles	286, no doubles	3	Using doubles will fail the project, floats OK!
	10	Bools	34, 158, 265-266, 27	4	
	11	sizeof *****			
	12	Variables 7 characters or less	36-42, 153-155, 245-249,		All variables <= 7 characters
	13	Scope ***** No Global Variables	none		
	14	Arithmetic operators	220, 294, 332, 454, 462, 5		*/ / - +
	15	Comments 20%+	306 comments	5	Model as pseudo code
	16	Named Constants	none		All Local, only Conversions/Physics/Math in Global area
	17	Programming Style ***** Emulate			Emulate style in book/in class repository
3	1	cin	68, 100, 113, 124, 174, 18		Inputs, press to continue
	2	Math Expression	220, 294, 332, 454, 462, 5		*/ / - +
	3	Mixing data types ****			
	4	Overflow/Underflow ****			
	5	Type Casting	24, 281, 294, 295	4	
	6	Multiple assignment *****			a=b=c=1
	7	Formatting output	309, setprecision	4	
	8	Strings	31, 44, 244	3	
	9	Math Library	cell: 261, 295	4	All libraries included have to be used
	10	Hand tracing *****	debugging		
4	1	Relational Operators	205, 209, 381, 386, 520, 5		>, >=, <, <=, ==, !=
	2	If	52, 69, 86, 115, 129	4	Independent If
	4	If-else	58-79, 182-230, 327	4	
	5	Nesting	58, 69, 110, 171, 18	4	
	6	If-else-If	554-638	4	
	7	Flags *****	85, 287, 231, 412, 545		
	8	Logical operators	69, 101, 115, 119, 1	4	&&, , !
	11	Validating user input	119, 180, 209, 386,	4	validate, confirm, do while
	13	Conditional Operator	121, 167-169, 222, 2	4	ternary operator
	14	Switch	218, 388, 432, 526,	4	
5	1	Increment/Decrement	365, 500, 647, 299,	4	
	2	While	287-675	4	
	5	Do-while	90-101, 104-125, 11	4	
	6	For loop	299, 304, 323, 443,	4	
	11	Files Input/output both	28-29, 56, 58, 61, 8	8	
	12	No breaks in loops *****	none		Failed Project if included
***** Not required to show			Total	100	

Flowcharts



Pseudocode

Original Pseudocode

```
//character creation menu
//move variables to store stats here
//bool create true
//variable to store input
//prompt to use saved character
//store input
//if y
//create = false
//check file
//if file exists
//file input
//else cout no file, returning to creation screen, create = true

//if create true
//character creation
//character name
//variable to store character name
//do
//prompt input
//store character name
//output name
//confirm name with do while y/n

//choose class
//variable to store class
//do
//prompt input, displaying info
//choose 0 for fighter or 1 for wizard
//0 Fighter 60 hp +2 con +3 str, +2 attack bonus, 16 armor (limits dex)
//1 Wizard 30 hp +3 int +2 dex, able to heal and cast mage armor
//do
//store class
//validate entry with if not 0 or 1 invalid and do while 0 or 1
//confirm entry with do while y/n

//set base stats based on class chosen
//variables to store stats
//if fighter 60 hp 16 ac +2 con +3 str +2 attack bonus total = str+2
//if wizard 30 hp 10 ac +3 int +2 dex attack bonus total = int
```

```

//choose stats
//variable to store point pool, stat choice, point allocation input, end bool
//int point pool, point allocation, points remaining = pool
//char choice
//bool end
//do
//prompt input to assign points, displaying info
//current hp, ac, stat bonuses, attack bonus, 5 points remaining
//1 con (1 pt = +5 hp)
//2 dex (1pt = +1 ac, if fighter display armor limitation)
//3 str (if fighter display 1pt = 1 attack dmg)
//4 int (if wizard display 1pt = 1 spell dmg)
//do
//store input char choice
//select stat switch char
//'1' con
//'2' dex
//'3' str
//'4' int
//'x' exit
//default invalid retry
//validate entry with do while not 1 to 4 or x
//if x
//confirm exit
//display points remaining, are you sure? cannot go back y/n
//if y end true
//else
//do
//prompt input of points from remaining-pool to remaining
//input points
//if invalid invalid try again
//validate entry with do while remaining-pool to points remaining
//update remaining = pool - input
//update stat += input switch char
//'1' con += input hp += 5*input
//'2' dex += input if wizard ac += input
//'3' str += input
//'4' int += input
//default "error"
//do while end is not true

//store character information in file

//enemy stats
//variables to store enemy stats
//ogre with fists 70 hp 15 ac 3 + 2d6 dmg per turn, 20 double special line, 1 dmg to

```

```

self

//output an ogre appears

//fight mechanics
//variables
    //bool isdead, loss
    //bars = hp/10
    //current hp
    //current enemy hp
    //action choice
    //damage
//while isDead is false or loss is false
    //display enemy health percentage bar
        //current bars = ceil current/10
        //float percent hp = cur/hp*100;
        //cout << "enemy HP: [";
        //for (i = 1; i <= current bars; i++) cout << |
        //for (i = 1; i <= bars - current bars; i++) cout << "."
        //cout << |
        //cout setprecision typecast percent

//display player current / hp

//enemy attack
//output the ogre reaches out
//roll d20 = rand() % 20
//preroll damage 2d6 +3
    //if roll + bonus > player ac
        //ternary 20 it hits vs it crits
        //ternary 20 dmg double
        //the orge does dmg
        //player hp adjustment
    //else
        //if 1 critical failure the orge did dmg to itself
        //enemy hp adjustment
        //else it missed

//action switch
//if fighter
    //var to store berserk = false, count = 0
    //if berserk = true count --, cout berserk active you have count rounds after this
one remaining
    //prompt action choice
    //variable to store input choice
    //display menu
/*

```



```

* 1 greatsword 1d12 + str
* 2 longbow 1d12 + str
* 3 dual handaxe 2d6 + str
* 4 dual scimitar 2d6 + str
* 5 no roll berserk double damage next three turns
*/
//store choice
//switch choice
  //1 greatsword : num = 1 sides = 12 name = greatsword
  //2 long bow damage roll = rand() % 12 +1;
  //3 dual handaxe :
  //4 dual scimitar damage roll = rand() % 6 +1 + rand() % 6 +1;
  //5 berserk berserk = true count += 3
//if 5 You spent your action to activate berserk. you have count rounds remaining
//else
  //output you raise your weapon name
  //pre roll damage
  //if berserk true double dmg
  //if count = 0 berserk = false
  //press enter to roll, you rolled a
  //roll d20 = rand() % 20
  //output you rolled a
  //if roll + bonus > enemy ac
    //ternary 20 you hit vs you crit
    //enter to roll for damage
    //output you rolled
    //tenary 20 dmg double
    //you did dmg
    //enemy hp adjustment
  //else
    //if 1 critical failure you did dmg to yourself
    //hp adjustment
    //else you missed

//if wizard
  //var to store armor = false, count = 0
  //if armor = true count --, cout armor active you have 18 ac for count rounds after
this one
  //prompt action choice
  //variable to store input choice
  //display menu
/*
  * 1 3d6 fireball wizard
  * 2 4d4 magic missiles wizard
  * 3 no roll heal 2d6 + int
  * 4 no roll mage armor 18 ac next three turns
*/

```

```

//store choice
//switch choice
  //1 fireball : num = 3 sides = 6 name = fireball
  //2 magic missiles 4d4
  //heal 2d6
  //4 mage armor armor = true count += 3
//if armor true ac = 18
//if 4 You spent your action to activate armor. your ac is 18. you have count rounds
remaining
//else if 3 you cast heal, roll to heal, adjust hp
  //else
  //output you cast spell name
  //pre roll damage
  //if count = 0 berserk = false
  //press enter to roll, you rolled a
  //roll d20 = rand() % 20
  //output you rolled a
  //if roll + bonus > enemy ac
    //ternary 20 you hit vs you crit
    //enter to roll for damage
    //output you rolled
    //tenary 20 dmg double
    //you did dmg
    //enemy hp adjustment
  //else
    //if 1 critical failure you did dmg to yourself
    //hp adjustment
    //else you missed

  //if enemy hp < 1 output slain, isdead true
  //if you hp < 1 putput slain, loss true
//if isdead is true
  //end screen congrats you won
//if loss true
  //end screen you died
//exit

```

Post-completion Pseudocode

```
import libraries
using std namespace
main function
declare char yn
declare in stream
declare out stream
declare string fname equal to "dndchr.dat"
declare bool create equal to true
declare int
con equal to 0
dex equal to 0
str equal to 0
inte equal to 0
hp equal to 0
ac equal to 10
bns equal to 0
declare char clas equal to 0
declare string name
output + "reload saved declare character? y/n" + new line
store input yn
if yn equals 'y' or yn equals 'y' then
    create equal to false
    open file
    if file not found equals false then
        in from file con dex str inte class hp ac bns
        getline for name
        output + new line + "you selected " + name + new line + "do you want to continue as "
        + name + "? y/n" + new line
        store input yn
        if yn not equal to 'y' and yn not equal to 'y' then
            create equal to true
        else
            create equal to true
            output + new line + "declare character file not found. returning to creation screen."
+ new line
        close file
    end if
end if
if create equals true then
    output + new line + "creating declare character." + new line + new line
    do
        output + "enter declare character name." + new line
        getline for name
```

```

    output + new line + "your declare character's name is " + name + ". is this correct?
y/n" + new line
    store input yn
    while yn not equal to 'y' and yn not equal to 'y'
    do
        output + new line + "choose your class." + new line + "select 0 for fighter and 1 for
wizard." + new line
        + "0: fighter. 60 hp. +2 con +3 str +2 attack bonus. armor - 16 ac limits dex. berserk
- double damage 3 rounds." + new line
        + "1: wizard. 30 hp. +3 declare int +2 dex. spells include healing and mage armor -
20 ac 3 rounds." + new line
    do
        store input class
        if class not equal to '0' and class not equal to '1' then
            output + new line + "invalid class. reenter." + new line
            while class not equal to '0' and class not equal to '1'
                output + "you selected " + if class equals '0' then if true "fighter" or if false "wizard"
then + "." + new line
                + "is this correct? y/n" + new line
            store input yn
            while yn not equal to 'y' and yn not equal to 'y'
            if class equals '0' then
                con add 3
                dex add 0
                str add 2
                inte add 0
                hp add 60
                ac add 6
                bns add 2 + str
            end if
            if class equals '1' then
                con add 0
                dex add 2
                str add 0
                inte add 3
                hp add 30
                ac add 0 + dex
                bns add 0 + inte
            end if
            declare int pool equal to 5
            pin
            remain equal to pool
            declare char cpin
            declare char ccho
            declare bool cend equal to false
            do
                output + new line + "hp: " + hp + " ac: " + ac + " attack bonus: " + bns + new line

```

```

+ "con: " + con + " dex: " + dex + " str: " + str + " int: " + inte + new line
+ "points remaining: " + remain + new line
+ "select stat to assign points." + new line
+ "1: con - 1 pt equal to +5 hp" + new line
+ "2: dex" + if class equals '0' then if true " - fighter ac is capped." or if false " - 1 pt
equal to +1 ac" then + new line
+ "3: str" + if class equals '0' then if true " - 1 pt equal to +1 attack bonus." or if false
"" then + new line
+ "4: int" + if class equals '0' then if true "" or if false " - 1 pt equal to +1 spell
bonus." then + new line
+ "press x to finish." + new line
do
  store input ccho
  if ccho not equal to '1' and ccho not equal to '2' and ccho not equal to '3' and ccho
not equal to '4' and ccho not equal to 'x' and ccho not equal to 'x' then
    output + new line + "invalid entry. reenter." + new line
  end if
  while ccho not equal to '1' and ccho not equal to '2' and ccho not equal to '3' and
ccho not equal to '4' and ccho not equal to 'x' and ccho not equal to 'x'
    if ccho equals 'x' or ccho equals 'x' then
      output + new line + "you have " + remain + " points remaining. are you sure you
want to exit? you cannot go back. y/n" + new line
      store input yn
      if yn equals 'y' or yn equals 'y' then
        cend equal to true
      else
        do
          do
            output + new line + "input number of points from " + 0 + " to " + remain +
new line
            store input cpin
            pin equal to cpin to int
            if pin is less than 0 or pin is greater than remain then
              output + new line + "invalid entry. reenter." + new line
            end if
            while pin is less than 0 or pin is greater than remain
              output + new line + pin + " points. is this correct? y/n" + new line
              store input yn
              while yn not equal to 'y' and yn not equal to 'y'
                remain - equal to pin
              switch through ccho
                for case '1': con add pin hp add 5 * pin break
                for case '2': dex add pin ac add if clas equals '1' then if true pin or if false 0
break
                for case '3': str add pin bns add if clas equals '0' then if true pin or if false 0
break
                for case '4': inte add pin bns add if clas equals '1' then if true pin or if false 0

```

```

break
    default: output + new line + "switch error" + new line
    end switch
end if
end if
while cend equals false
open file
    store in out + con + new line + dex + new line + str + new line + inte + new line + clas +
new line + hp + new line + ac + new line + bns + new line + name
close file
declare string ename equal to "ogre"
declare int ehp equal to 70
eac equal to 15
ebns equal to 6
en equal to 2
ed equal to 6
output + new line + "ready?" + new line
store input yn
output + new line + "great." + new line
    + "..." + new line
    + "an " + ename + " appears."
output + new line + new line + "press any key to continue." + new line
store input yn
declare bool isd equal to false
ised equal to false
declare int chp equal to hp
r20
r20b
rdmg
dmg
n
d
turn equal to 1
pupcnt equal to 0
declare char act
declare bool pup equal to false
declare int cehp equal to ehp
ebar equal to ceil of ehp/10
cebar
er20
er20b
erdmg
declare float pct
while isd equals false and ised equals false
    output + "_____ "
    pct equal to cehp/ehp * 100.0f
    cebar equal to ceil of cehp/10

```

```

output + new line + "enemy hp: ["
for loop declare int i equal to 1 if i is less than or equal to cbar as i increases
  output + "|"
for loop declare int i equal to 1 if i is less than or equal to ebar - cbar as i increases
  output + "_"
output + "]" + fixed + showpoint + setprecision for 2 then + pct + " %" + new line
output + "player hp: " + chp + "/" + hp + new line
output + "turn: " + turn + new line
output + new line + "the " + ename + " reaches out." + new line
er20 equal to rand%20 + 1
er20b equal to er20 + ebns
erdmg equal to ebns
for loop declare int i equal to 1 if i is less than or equal to en as i increases
  erdmg add rand%ed
  if er20b is greater than or equal to ac then
    output + new line + "the " + ename + " pummels " + name + " and " + if er20 equals
20 then if true "crits " or if false "hits. " then
    erdmg equal to er20 if equals 20 then if true erdmg*2 or if false erdmg
    output + name + " takes " + erdmg + " damage." + new line
    chp subtract erdmg
  else
    if er20 equals 1 then
      output + new line + "critical failure. the " + ename + " trips and hits itself losing "
+ erdmg + " hp." + new line
      cehp subtract erdmg
    else
      output + new line + "the " + ename + " misses. it roars angrily." + new line
    end if
  end if
output + new line + "your turn. press any key to continue." + new line
store input yn
if class equals '0' then
  if pup equals true then
    pupcnt decreases
    output + new line + "berserk active. you have double damage for " + pupcnt + "
rounds after this one remaining." + new line
    output + new line + "choose an action." + new line
    output + "1: greatsword 1d12 + " + bns + new line
    + "2: longbow 1d12 + " + bns + new line
    + "3: dual handaxe 2d6 + " + bns + new line
    + "4: dual scimitar 2d6 + " + bns + new line
    + "5: berserk - double damage for the next three turns" + new line
  do
    store input act
    if act is less than '1' or act is greater than '5' then
      output + new line + "invalid entry. reenter." + new line
    end if

```

```

while act is less than '1' or act is greater than '5'
switch through act
  for case '1':
    n equal to 1
    d equal to 12
    output + new line + "you raise your greatsword." + new line
    break
  for case '2':
    n equal to 1
    d equal to 12
    output + new line + "you aim your longbow." + new line
    break
  for case '3':
    n equal to 2
    d equal to 6
    output + new line + "you spin your handaxes." + new line
    break
  for case '4':
    n equal to 2
    d equal to 6
    output + new line + "you flash your scimitars." + new line
    break
  for case '5':
    pup equal to true
    popcnt equal to 3
    output + new line + "your eyes flash as you activate berserk." + new line
      + "you spent your action. you have " + popcnt + " rounds remaining."
    break
  default:
    output + "switch error"
end switch
if act is greater than or equal to '1' and act is less than or equal to '4' then
  output + new line + "press any key to roll the d20." + new line
  store input yn
  r20 equal to rand%20 + 1
  r20b equal to r20 + bns
  output + new line + "you rolled a natural " + r20
  switch through r20
    for case 1:
      output + ". critical failure."
      break
    for case 20:
      output + ". critical hit - double damage. attack bonus: " + bns + ". your total
is " + r20b + "."
      break
    default:
      output + ". attack bonus: " + bns + ". your total is " + r20b + "."

```



```

end switch
output + new line + "press any key to roll for damage - " + n + "d" + d + new line
store input yn
for loop declare int i equal to 1 if i is less than or equal to n as i increases
  rdmg equal to rand%d +1
  output + "you rolled a " + rdmg + ". "
  dmg add rdmg
  dmg add bns
end for loop
if pup equals true then
  dmg *equal to 2
end if
if pupcnt equals 0 then
  pup equal to false
end if
if r20b is greater than or equal to eac then
  output + new line + "success. " + name + " lands a blow on the " + ename + "
and " + if r20 equals 20 then if true "crits. or if false "hits." then + new line
  output + "the " + ename + " takes " + dmg + " damage." + new line
  cehp subtract dmg
else
  if r20 equals 1 then
    output + new line + "critical failure. " + name + " trips and stabs himself
losing " + dmg + " hp." + new line
    chp subtract dmg
  else
    output + new line + name + " leaps forward and misses. you do no damage."
+ new line
  end if
end if
end if
else if class equals '1' then
  if pup equals true then
    ac equal to 20
    output + new line + "mage armor active. you have " + ac + " ac for " + pupcnt + "
rounds after this one remaining." + new line
    pupcnt decreases
  end if
output + new line + "choose an action." + new line
  output + "1: fireball 3d6 + " + bns + new line
  + "2: magic missiles 4d4 + " + bns + new line
  + "3: heal 2d6 + " + bns + new line
  + "4: mage armor - 20 ac for the next three turns" + new line
store input act
do
  store input act
  output + new line + "invalid entry. reenter." + new line

```

```

while act is less than '1' or act is greater than '4'
switch through act
  for case '1':
    n equal to 3
    d equal to 6
    output + new line + "you cast fireball. it launches across the field." + new
line
    break
  for case '2':
    n equal to 4
    d equal to 4
    output + new line + "you fire magic missiles." + new line
    break
  for case '3':
    n equal to 2
    d equal to 6
    output + new line + "you cast a healing spell." + new line
    break
  for case '4':
    pup equal to true
    pupcnt equal to 3
    output + new line + "the air shimmers as you activate mage armor." + new
line
    + "you spent your action. you have " + pupcnt + " rounds remaining."
    break
  default:
    output + "switch error"
end switch
if act equals '1' or act equals '2' then
  output + new line + "press any key to roll the d20." + new line
  store input yn
  r20 equal to rand%20 + 1
  r20b equal to r20 + bns
  output + new line + "you rolled a natural " + r20
  switch through r20
    for case 1:
      output + ". critical failure."
      break
    for case 20:
      output + ". critical hit - double damage. spell bonus: " + bns + ". your total
is " + r20b + "."
      break
    default:
      output + ". spell bonus: " + bns + ". your total is " + r20b + "."
  end switch
  output + new line + "press any key to roll for damage - " + n + "d" + d + new
line

```

```

store input yn
for loop declare int i equal to 1 if i is less than or equal to n as i increases
  rdmg equal to rand%d +1
  output + "you rolled a " + rdmg + ". "
  dmg add rdmg
  dmg add bns
end for loop
dmg equal to r20 if equals 20 then if true dmg*2 or if false dmg
output + new line + dmg + " damage total." + new line
if r20b is greater than or equal to eac then
  output + new line + "success. " + name + "'s spell lands a blow on the " +
ename + " and " + if r20 equals 20 then if true "crits. or if false "hits." then + new line
  output + "the " + ename + " takes " + dmg + " damage." + new line
  chp subtract dmg
else
  if r20 equals 1 then
    output + new line + "critical failure. " + name + " loses concentration and
turns a toe into a frog losing " + dmg + " hp." + new line
    chp subtract dmg
  else
    output + new line + name + "'s spell misfires. you do no damage." + new
line
  end if
end if
else if act equals '3' then
  output + new line + "press any key to heal " + n + "d" + d + " health points. " +
new line
  store input yn
  for loop declare int i equal to 1 if i is less than or equal to n i increases
    rdmg equal to rand%d +1
    output + "you rolled a " + rdmg + ". "
    dmg add rdmg
    dmg add bns
  end for loop
  output + "spell bonus: " + bns + ". "
  output + new line + "you regained " + dmg + " hp total." + new line
  chp add dmg
  if chp is greater than hp then
    chp equal to hp
  end if
  output + new line + "press any key to continue." + new line
  store input yn
end if
end if
end if
turn increases
reset dmg equal to 0

```

```
if cehp is less than 1 then
    output + new line + "enemy hp: " + cehp + ". success." + new line
    + new line + "the " + ename + " has been slain." + new line
    + "your name " + name + " will be memorialized in ballads far and wide." + new
line
    isd equal to true
end if
if chp is less than 1 then
    output + new line + "you died" + new line
    output + new line + "you " + name + " have been slain
    isd equal to true
if chp is less than 1 and cehp is less than 1 then
    output + name + " ended the " + ename + "'s reign of terror at the cost of their own
life."
    + new line
exit main
```

Code

Project 1 Final Version v8

```
1 /*
2 * File:  main.cpp
3 * Author: knguyen
4 * Purpose: Final Version of text based simplified DnD game.
5 * Minor experience adjustments. Polishing code. Complete game.
6 */
7 //sys lib
8 #include <iostream>
9 #include <fstream> //File I/O
10 #include <iomanip> //Format
11 #include <string> //String
12 #include <cstdlib> //Rand function
13 #include <ctime> //Time to set random function seed
14 #include <cmath> //For rounding capability
15
16 using namespace std;
17
18 //later add functions
19 //prototypes
20
21 int main(int argc, char** argv)
22 {
23     //Seed rand with the current time
24     srand(static_cast<unsigned int>(time(0)));
25
26     char yn; //variable to store yes or no input
27
28     ifstream in; //declare in stream
29     ofstream out; //declare out stream
30
31     string fname = "dndchr.dat"; //variable to store file name
32
33     //character creation
34     bool create = true; //bool for creation screen
35     //variables to store stats
36     int con = 0, //constitution
37         dex = 0, //dexterity
38         str = 0, //strength
39         inte = 0, //intelligence
```

```
40     hp = 0, //health points
41     ac = 10, //armor class
42     bns = 0; //attack bonus
43 char clas = 0; //class
44 string name; //name
45
46 //load saved character
47 //prompt to use saved character
48 cout << "Reload saved character? y/n" << endl;
49 //store input
50 cin >> yn;
51 //conditional to load saved character
52 if (yn == 'y' || yn == 'Y')
53 {
54     create = false; //toggle create screen
55     //open file
56     in.open(fname, ios::in);
57     //check for file
58     if (in.fail() == false) //if file can be found
59     {
60         //store input from file
61         in >> con >> dex >> str >> inte >> clas >> hp >> ac >> bns; //int values
62         in.ignore(); //ignore endl
63         getline(in, name); //getline for string
64         //verify choice
65         cout << endl << "You selected " << name << endl
66             << "Do you want to continue as " << name << "? y/n" << endl;
67         //store input
68         cin >> yn;
69         if (yn != 'y' && yn != 'Y') //if not yes
70         {
71             create = true; //toggle character creation
72         }
73     }
74     else //if file cannot be found
75     {
76         create = true; //toggle character creation
77         //inform player
78         cout << endl << "Character file not found. Returning to creation screen." <<
endl;
79     }
80     //close file
81     in.close();
82 }
83
84 //character creation screen
85 if (create == true)
```

```
86  {
87    //inform player
88    cout << endl << "Creating character." << endl << endl;
89    //character name
90    do
91    {
92        //prompt input
93        cout << "Enter character name." << endl;
94        //store character name
95        cin.ignore(); //ignore last endl
96        getline(cin, name); //store name from cin
97        //confirm name
98        cout << endl << "Your character's name is " << name << ". Is this correct? y/n" <<
endl;
99        //store input
100        cin >> yn;
101    } while (yn != 'y' && yn != 'Y'); //confirm entry with do while
102
103    //choose class
104    do
105    {
106        //prompt input
107        cout << endl << "Choose your class." << endl << "Select 0 for Fighter and 1 for
Wizard." << endl
108            << "0: Fighter. 60 HP. +2 con +3 str, +2 attack bonus. Armor - 16 AC, limits
dex. Berserk - double damage, 3 rounds." << endl
109            << "1: Wizard. 30 HP. +3 int +2 dex. Spells include Healing, and Mage
Armor - 20 AC, 3 rounds." << endl;
110        do
111        {
112            //store class
113            cin >> clas;
114            //output invalid message
115            if (clas != '0' && clas != '1')
116            {
117                cout << endl << "Invalid class. Reenter." << endl;
118            }
119        } while (clas != '0' && clas != '1'); //validate entry with do while
120        //confirm entry
121        cout << "You selected " << ( (clas == '0')? "fighter" : "wizard" ) << ". " << endl
122            << "Is this correct? y/n" << endl;
123        //store input
124        cin >> yn;
125    } while (yn != 'y' && yn != 'Y'); //confirm entry with do while
126
127    //set base stats based on class chosen
128    //set fighter stats
```

```

129     if (clas == '0')
130     {
131         con += 3, //constitution
132         dex += 0, //dexterity
133         str += 2, //strength
134         inte += 0, //intelligence
135         hp += 60, //health points
136         ac += 6, //armor class
137         bns += 2 + str; //attack bonus
138     }
139     //set wizard stats
140     if (clas == '1')
141     {
142         con += 0, //constitution
143         dex += 2, //dexterity
144         str += 0, //strength
145         inte += 3, //intelligence
146         hp += 30, //health points
147         ac += 0 + dex, //armor class
148         bns += 0 + inte; //attack bonus
149     }
150
151     //choose stats
152     //variable to store point pool, stat choice, point allocation input, end bool
153     int pool = 5, //point pool
154         pin, //store points entered
155         remain = pool; //remaining points
156     char cpin; //store points earned as char to avoid cin error
157     char ccho; //to store stat selection choice
158     bool cend = false; //to end stat allocation
159     do //do until exit confirmed
160     {
161         //prompt input to assign points, displaying info
162         cout << endl << "HP: " << hp << " AC: " << ac << " Attack bonus: " << bns <<
endl
163         << "con: " << con << " dex: " << dex << " str: " << str << " int: " << inte <<
endl
164         << "Points remaining: " << remain << endl
165         << "Select stat to assign points." << endl
166         << "1: con - 1 pt = +5 HP" << endl
167         << "2: dex" << ( (clas == '0')? " - Fighter AC is capped." : " - 1 pt = +1 AC" ) <<
endl
168         << "3: str" << ( (clas == '0')? " - 1 pt = +1 attack bonus." : "" ) << endl
169         << "4: int" << ( (clas == '0')? "" : " - 1 pt = +1 spell bonus." ) << endl
170         << "Press x to finish." << endl;
171     do
172     {

```



```
173         //store input char choice
174         cin >> ccho;
175         //output invalid message
176         if (ccho != '1' && ccho != '2' && ccho != '3' && ccho != '4' && ccho != 'x' &&
ccho != 'X')
177         {
178             cout << endl << "Invalid entry. Reenter." << endl;
179         }
180     } while (ccho != '1' && ccho != '2' && ccho != '3' && ccho != '4' && ccho != 'x' &&
ccho != 'X'); //validate with do while
181
182     if (ccho == 'x' || ccho == 'X') //if exit chosen
183     {
184         //confirm exit
185         cout << endl << "You have " << remain << " points remaining. Are you sure
you want to exit? You cannot go back. y/n" << endl;
186         //store input
187         cin >> yn;
188         if (yn == 'y' || yn == 'Y') //if yes
189         {
190             cend = true; //toggle exit, stat screen off
191         }
192     }
193     else
194     {
195         do //confirm entry
196         {
197             do //validate input
198             {
199                 //prompt input
200                 cout << endl << "Input number of points from " << 0 << " to " << remain
<< endl;
201                 //store char input
202                 cin >> cpin;
203                 pin = cpin - '0'; //clean and convert char to int
204                 //output invalid message if pin not in range
205                 if ( (pin < 0 || pin > remain) )
206                 {
207                     cout << endl << "Invalid entry. Reenter." << endl;
208                 }
209             } while ( pin < 0 || pin > remain ); //validate input with do while
210             //confirm entry
211             cout << endl << pin << " points. Is this correct? y/n" << endl;
212             //store input
213             cin >> yn;
214         } while (yn != 'y' && yn != 'Y'); //confirm entry with do while
215         //update remaining points
```

```

216         remain -= pin;
217         //switch to update stats
218         switch (ccho)
219         {
220             case '1': con += pin; hp += 5 * pin; //update con and hp
221                 break;
222             case '2': dex += pin; ac += ( (clas == '1')? pin : 0 ); //update dex and ac if
wizard
223                 break;
224             case '3': str += pin; bns += ( (clas == '0')? pin : 0 ); //update str and bonus if
fighter
225                 break;
226             case '4': inte += pin; bns += ( (clas == '1')? pin : 0 ); //update str and bonus
if wizard
227                 break;
228             default: cout << endl << "switch error" << endl;
229         }
230     }
231 } while (cend == false); //do while stat screen is toggled on
232
233 //store character information in file
234 //open file
235 out.open(fname,ios::out);
236 //output into file
237 out << con << endl << dex << endl << str << endl << inte << endl << clas << endl
<< hp << endl << ac << endl << bns << endl << name;
238 //close file
239 out.close();
240 }
241
242 //enemy stats
243 //variables to store enemy stats
244 string ename = "ogre";
245 int ehp = 70, //health points
246     eac = 15, //armor class
247     ebns = 6, //attack bonus
248     en = 2, //number of dice
249     ed = 6; //dice sides
250
251 //output beginning of fight
252 cout << endl << "Ready?" << endl;
253 cin >> yn;
254 cout << endl << "Great." << endl
255     << "..." << endl
256     << "An " << ename << " appears.";
257
258 //prompt continue

```

```

259 cout << endl << endl << "Press any key to continue." << endl;
260 //enter to continue
261 cin >> yn;
262
263 //fight mechanics
264 //variables to keep track of battle
265 bool isD = false, //is dead
266     iseD = false; //is enemy dead
267 //variables for player
268 int chp = hp, //current hp
269     r20, //store roll for attack
270     r20b, //store attack roll plus bonus
271     rdmg, //store roll for damage
272     dmg, //store total damage
273     n, //number of dice
274     d, //number of sides
275     turn = 1, //store turn
276     pupcnt = 0; //store powerup count
277 char act; //action choice
278 bool pup = false; //power up bool
279 //enemy variables
280 int cehp = ehq, //current enemy hp
281     ebar = ceil(static_cast<float>(ehp)/10), //number of health bars from enemy hp,
rounding up decimal
282     cebar, //store number of current health bars
283     er20, //store roll for attack
284     er20b, //store attack roll plus bonus
285     erdmg; //store roll for damage
286 float pct; //store percent
287 while (isD == false && iseD == false) //while both are alive
288 {
289     //Mark new turn
290     cout << "_____";
291
292     //display enemy health percentage bar
293     //store calculations
294     pct = static_cast<float>(cehp)/static_cast<float>(ehp) * 100.0f; // health
percentage
295     cebar = ceil(static_cast<float>(cehp)/10); //current bars, rounding up decimal
296     //start output
297     cout << endl << "Enemy HP: [";
298     //loop through current health bars
299     for (int i = 1; i <= cebar; i++)
300     {
301         cout << "|";
302     }
303     //loop through depleted health bars

```

```

304     for (int i = 1; i <= ( ebar - cebar ); i++)
305     {
306         cout << "_";
307     }
308     cout << "]" //end bar
309         << fixed << showpoint << setprecision(2) << pct << " %" << endl; //output
percent at 2 points
310
311     //display player current / hp
312     cout << "Player HP: " << chp << "/" << hp << endl;
313
314     //display turn
315     cout << "Turn: " << turn << endl;
316
317     //enemy attack
318     //output attack
319     cout << endl << "The " << ename << " reaches out." << endl;
320     er20 = rand()%20 +1; //roll for attack
321     er20b = er20 + ebns; //attack roll plus bonus
322     erdmg = ebns; //roll for damage + bonus
323     for (int i = 1; i <= en; i++) //roll for damage for loop number of dice
324     {
325         erdmg += rand()%ed; // add roll
326     }
327     if (er20b >= ac) //if roll > ac, hits
328     {
329         //output attack line, if natural 20 critical hit
330         cout << endl << "The " << ename << " pummels " << name << " and " << ( (er20
== 20)? "crits! " : "hits. ");
331         //double damage if critical
332         erdmg = (er20 == 20)? erdmg*2 : erdmg;
333         //output damage done
334         cout << name << " takes " << erdmg << " damage." << endl;
335         //update player hp
336         chp -= erdmg;
337     }
338     else
339     {
340         if (er20 == 1) //if natural roll is 1, critical failure
341         {
342             //output critical failure
343             cout << endl << "Critical failure. The " << ename << " trips and hits itself,
losing " << erdmg << " HP." << endl;
344             //update enemy hp
345             cehp -= erdmg;
346         }
347         else //else normal miss

```

```

348     {
349         //output dialogue
350         cout << endl << "The " << ename << " misses. It roars angrily." << endl;
351     }
352 }
353
354 //player attack
355 //begin output
356 cout << endl << "Your turn. Press any key to continue." << endl;
357 //enter to continue
358 cin >> yn;
359 //action switch
360 //if fighter
361 if (clas == '0')
362 {
363     if (pup == true) //if powerup is active
364     {
365         pupcnt--; //decrease count per round
366         cout << endl << "Berserk active. You have double damage for " << pupcnt <<
" rounds after this one remaining." << endl; //inform player
367     }
368     //prompt action choice
369     cout << endl << "Choose an action." << endl;
370     //display menu
371     cout << "1: greatsword 1d12 + " << bns << endl
372         << "2: longbow 1d12 + " << bns << endl
373         << "3: dual handaxe 2d6 + " << bns << endl
374         << "4: dual scimitar 2d6 + " << bns << endl
375         << "5: Berserk - double damage for the next three turns" << endl;
376     //store choice
377     do
378     {
379         //store action
380         cin >> act;
381         if (act < '1' || act > '5')
382         {
383             //invalid message
384             cout << endl << "Invalid entry. Reenter." << endl;
385         }
386     } while (act < '1' || act > '5'); //validate input
387     //switch choice
388     switch (act)
389     {
390         //weapons, number of dice and sides
391         case '1':
392             n = 1;
393             d = 12;

```

```

394         cout << endl << "You raise your greatsword." << endl;
395         break;
396     case '2':
397         n = 1;
398         d = 12;
399         cout << endl << "You aim your longbow." << endl;
400         break;
401     case '3':
402         n = 2;
403         d = 6;
404         cout << endl << "You spin your handaxes." << endl;
405         break;
406     case '4':
407         n = 2;
408         d = 6;
409         cout << endl << "You flash your scimitars." << endl;
410         break;
411     case '5': //berserk
412         pup = true; //toggle powerup on
413         pupcnt = 3; //set count for rounds remaining
414         cout << endl << "Your eyes flash as you activate Berserk." << endl
415             << "You spent your action. You have " << pupcnt << " rounds
remaining.";
416         break;
417     default: cout << "switch error";
418 }
419 if (act >= '1' && act <= '4') //if attack action
420 {
421     //roll to attack
422     //prompt roll
423     cout << endl << "Press any key to roll the d20." << endl;
424     //enter for "roll"
425     cin >> yn;
426     //roll for attack
427     r20 = rand()%20 + 1;
428     r20b = r20 + bns; //attack roll plus bonus
429     //output roll
430     cout << endl << "You rolled a natural " << r20;
431     //switch output depending on natural roll
432     switch (r20)
433     {
434         case 1: cout << ". Critical failure."; break;
435         case 20: cout << ". Critical hit - Double damage. Attack bonus: " << bns <<
". Your total is " << r20b << "."; break;
436         default: cout << ". Attack bonus: " << bns << ". Your total is " << r20b << ".";
437     }
438

```

```

439      //prompt damage roll
440      cout << endl << "Press any key to roll for damage - " << n << "d" << d <<
endl;
441      //enter for "roll"
442      cin >> yn;
443      for (int i = 1; i <= n; i++) //roll for damage for loop number of dice
444      {
445          rdmg = rand()%d +1; //damage roll
446          cout << "You rolled a " << rdmg << ". "; //output roll
447          dmg += rdmg; //add to total damage
448      }
449      dmg += bns; //total roll for damage + bonus
450      cout << "Attack bonus: " << bns << ". "; //output bonus
451      if (pup == true) //if berserk is on
452      {
453          cout << "Berserk: Double damage."; //output berserk notification
454          dmg *= 2; //double damage
455      }
456      //if count reaches zero
457      if (pupcnt == 0)
458      {
459          pup = false; //toggle berserk off
460      }
461      //double damage if critical
462      dmg = (r20 == 20)? dmg*2 : dmg;
463      //output total damage
464      cout << endl << dmg << " damage total." << endl;
465
466      if (r20b >= eac) //if roll meets or exceeds ac, hits
467      {
468          //output attack line, if natural 20 critical hit
469          cout << endl << "Success. " << name << " lands a blow on the " << ename
<< " and " << ( (r20 == 20)? "crits!" : "hits.") << endl;
470          //output damage done
471          cout << "The " << ename << " takes " << dmg << " damage." << endl;
472          //update player hp
473          cehp -= dmg;
474      }
475      else
476      {
477          if (r20 == 1) //if natural roll is 1, critical failure
478          {
479              //output critical failure
480              cout << endl << "Critical failure. " << name << " trips and stabs themself,
losing " << dmg << " HP." << endl;
481              //update enemy hp
482              chp -= dmg;

```

```
483         }
484         else //else normal miss
485         {
486             //output dialogue
487             cout << endl << name << " leaps forward and misses. You do no
damage." << endl;
488         }
489     }
490 }
491 }
492
493 //if wizard
494 if (clas == '1')
495 {
496     if (pup == true) //if powerup is active
497     {
498         ac = 20; //set armor class to 20
499         cout << endl << "Mage Armor active. You have " << ac << " AC for " <<
pupcnt << " rounds after this one remaining." << endl; //inform player
500         pupcnt--; //decrease count per round
501         if (pupcnt == 0) //if count reaches zero
502         {
503             pup = false; //toggle armor off
504         }
505     }
506
507     //prompt action choice
508     cout << endl << "Choose an action." << endl;
509     //display menu
510     cout << "1: fireball 3d6 + " << bns << endl
511         << "2: magic missiles 4d4 + " << bns << endl
512         << "3: heal 2d6 + " << bns << endl
513         << "4: mage armor - 20 AC for the next three turns" << endl;
514     //store choice
515     do
516     {
517         //store action
518         cin >> act;
519         //output invalid message
520         if (act < '1' || act > '4')
521         {
522             cout << endl << "Invalid entry. Reenter." << endl;
523         }
524     } while (act < '1' || act > '4'); //validate input
525     //switch choice
526     switch (act)
527     {
```



```

528      //weapons, number of dice and sides
529      case '1':
530          n = 3;
531          d = 6;
532          cout << endl << "You cast fireball. It launches across the field." << endl;
533          break;
534      case '2':
535          n = 4;
536          d = 4;
537          cout << endl << "You fire magic missiles." << endl;
538          break;
539      case '3':
540          n = 2;
541          d = 6;
542          cout << endl << "You cast a healing spell." << endl;
543          break;
544      case '4': //mage armor
545          pup = true; //toggle powerup on
546          pupcnt = 3; //set count for rounds remaining
547          cout << endl << "The air shimmers as you activate Mage Armor." << endl
548              << "You spent your action. You have " << pupcnt << " rounds
remaining.";
549          break;
550      default: cout << "switch error";
551  }
552
553  //execute actions with rolls
554  if (act == '1' || act == '2')
555  {
556      //roll to attack
557      //prompt roll
558      cout << endl << "Press any key to roll the d20." << endl;
559      //enter for "roll"
560      cin >> yn;
561      //roll for attack
562      r20 = rand()%20 + 1;
563      r20b = r20 + bns; //attack roll plus bonus
564      //output roll
565      cout << endl << "You rolled a natural " << r20;
566      //switch output depending on natural roll
567      switch (r20)
568      {
569          case 1: cout << ". Critical failure."; break;
570          case 20: cout << ". Critical hit - Double damage. Spell bonus: " << bns << ".
Your total is " << r20b << "."; break;
571          default: cout << ". Spell bonus: " << bns << ". Your total is " << r20b << ".";
572      }

```

```

573
574     //prompt damage roll
575     cout << endl << "Press any key to roll for damage - " << n << "d" << d <<
endl;
576     //enter for "roll"
577     cin >> yn;
578     for (int i = 1; i <= n; i++) //roll for damage for loop number of dice
579     {
580         rdmg = rand()%d + 1; //damage roll
581         cout << "You rolled a " << rdmg << ". "; //output damage roll
582         dmg += rdmg; //add to total damage
583     }
584     dmg += bns; //total roll for damage + bonus
585     cout << "Spell bonus: " << bns << ". "; //output bonus
586     //double damage if critical
587     dmg = (r20 == 20)? dmg*2 : dmg;
588     //output total damage
589     cout << endl << dmg << " damage total." << endl;
590
591     if (r20b >= eac) //if roll meets or exceeds ac, hits
592     {
593         //output attack line, if natural 20 critical hit
594         cout << endl << "Success. " << name << "'s spell lands a blow on the " <<
ename << " and " << ( r20 == 20)? "crits!" : "hits." << endl;
595         //output damage done
596         cout << "The " << ename << " takes " << dmg << " damage." << endl;
597         //update player hp
598         cehp -= dmg;
599     }
600     else
601     {
602         if (r20 == 1) //if natural roll is 1, critical failure
603         {
604             //output critical failure
605             cout << endl << "Critical failure. " << name << " loses concentration and
turns a toe into a frog, losing " << dmg << " HP." << endl;
606             //update enemy hp
607             chp -= dmg;
608         }
609         else //else normal miss
610         {
611             //output dialogue
612             cout << endl << name << "'s spell misfires. You do no damage." << endl;
613         }
614     }
615 }
616 else if (act == '3') //else if 3 you cast heal, roll to heal, adjust hp

```

```

617     {
618         //prompt healing roll
619         cout << endl << "Press any key to heal " << n << "d" << d << " health points. "
<< endl;
620         //enter for "roll"
621         cin >> yn;
622         for (int i = 1; i <= n; i++) //roll for healing points for loop number of dice
623         {
624             rdmg = rand()%d +1; //healing roll using same damage variable
625             cout << "You rolled a " << rdmg << ". "; //output roll
626             dmg += rdmg; //add to total healing points
627         }
628         dmg += bns; //total roll for healing points + bonus
629         cout << "Spell bonus: " << bns << ". "; //output bonus
630         //output total healing points
631         cout << endl << "You regained " << dmg << " HP total." << endl;
632         //update hp
633         chp += dmg;
634         if (chp > hp) //if healing points lead to current hp exceeding max hp
635         {
636             chp = hp; //set equal to max hp
637         }
638     }
639 }
640
641 //prompt continue
642 cout << endl << "Press any key to continue." << endl;
643 //enter to continue
644 cin >> yn;
645
646 //update turn
647 turn ++;
648 //reset damage
649 dmg = 0;
650
651 //if enemy dead
652 if (cehp < 1)
653 {
654     //output results
655     cout << endl << "Enemy HP: " << ce hp << ". Success!" << endl
<< endl << "The " << ename << " has been slain." << endl
<< "Your name, " << name << ", will be memorialized in ballads far and
wide." << endl;
658     iseD = true; //toggle death end screen
659 }
660 //if you are dead
661 if (chp < 1)

```

```
662     {
663         //output results
664         cout << endl << "YOU DIED" << endl;
665         cout << endl << "You, " << name << ", have been slain. Your final act was
injuring the " << ename << "." << endl
666         << "However, your journey has come to an end." << endl << chp << " HP."
<< endl;
667         isD = true; //toggle death end screen
668     }
669     //if both dead
670     if (chp < 1 && cehp < 1)
671     {
672         //output results
673         cout << name << " ended the " << ename << "'s reign of terror, at the cost of
their own life." << endl;
674     }
675 }
676 //exit
677 return 0;
678 }
```

Conclusion

This project for text based DnD relies on input and output, switching cases between choices, as well as do while loops to verify inputs, if and else statements including nested statements to regulate mechanisms, and a dice rolling mechanic based on random numbers. In the peripherals are file input and output, for loops for repetitive tasks, varying variable types, and varying operations and operators.

There is room for heavy improvement and expansion of the project involving functions, arrays, and sorting.