# Assignment Description

You are working on character creation for a new RPG. You will create a character struct based on the specification listed below:

# GitHub URL (optional)

[https://github.com/wesleyhixon/Programming-Assignments/tree/9dbf68cbf0c8276421ca6534b2963de3e6147eaf/M07%20Programming%20Assignment%201](https://github.com/wesleyhixon/Programming-Assignments/tree/9dbf68cbf0c8276421ca6534b2963de3e6147eaf/M07 Programming Assignment 1)

# Readme Documentation

Input Information: Input is a character name followed by the character class

Output Information: Output is the characters created with their name and class stats

# Flowchart Screen Shots (optional)

Screen shot(s) here

# UML and Use Case Diagrams (optional)

Screen shot(s) here

# Source Code of All files (.h, .cpp)

#include *<iostream>*

#include *<iomanip>*

**using** **namespace** **std**;

*/\**

*Program Name: RPG Characters*

*Author: Wesley Hixon*

*Date Last Updated: 07/16/2024*

*Purpose: Character creator for an RPG game using structs and enum*

*\*/*

*// Declaring enum and struct globally*

**enum** **characterClassType**{ FIGHT, ROGUE, MAGIC, CLERIC };

**struct** **characterType**{

string name;

double hp;

double mp;

int strength;

int dexterity;

int intellegence;

int speed;

int endurance;

int faith;

characterClassType characterClass;

};

*// Function prototypes*

void displayCharacter(characterType theCharacter); *// Display to the console the stats and info about a single character*

void createCharacter(characterType& character); *// Character creation function*

bool continuePrompt(); *// Prompt to continue creating characters*

int main(){

characterType playerCharacters[10]; *// Player characters array*

int characterNum = 0; *// Characters num for controlling array*

bool running = true;

char continueChoice;

**while**(running){

**if**(characterNum <= 9){

createCharacter(playerCharacters[characterNum]); *// Creating new character*

characterNum++; *// Iterating to next character*

running = continuePrompt(); *// Prompting if user would like to continue*

cin.ignore(); *// This is here to clear '\n' for getline() for the next character name*

} *// Otherwise getline() reads the '\n' in the buffer and doesn't take input*

**else**{

cout << "You have entered the max number of characters. The program will now exit." << endl;

running = false; *// In case of 10 characters entered*

}

}

cout << endl << "Here are the characters you created: " << endl;

**for**(int i = 0; i < characterNum; i++){ *// Using for loop to output every character*

displayCharacter(playerCharacters[i]);

}

**return** 0;

}

*// Display to the console the stats and info about a single character*

void displayCharacter(characterType theCharacter){

string classString;

**switch**(theCharacter.characterClass){ *// Getting string for class based on enum*

**case** FIGHT:

classString = "Fighter";

**break**;

**case** ROGUE:

classString = "Rogue";

**break**;

**case** MAGIC:

classString = "Magician";

**break**;

**case** CLERIC:

classString = "Cleric";

**break**;

}

cout << endl << theCharacter.name << " the " << classString << "'s stats:" << endl; *// Outputting stats*

cout << "HP: " << theCharacter.hp << endl;

cout << "MP: " << theCharacter.mp << endl;

cout << "Strength: " << theCharacter.strength << endl;

cout << "Dexterity: " << theCharacter.dexterity << endl;

cout << "Intelligence: " << theCharacter.intellegence << endl;

cout << "Speed: " << theCharacter.speed << endl;

cout << "Endurance: " << theCharacter.endurance << endl;

cout << "Faith: " << theCharacter.faith << endl;

}

*// Character creation function*

void createCharacter(characterType& character){

characterType defaultClasses[4]; *// These are declarations for the default classes*

defaultClasses[FIGHT] = (characterType){ *// Fighter class*

.hp = 30,

.mp = 0,

.strength = 16,

.dexterity = 10,

.intellegence = 5,

.speed = 8,

.endurance = 15,

.faith = 5,

.characterClass = FIGHT

};

defaultClasses[ROGUE] = (characterType){ *// Rogue class*

.hp = 20,

.mp = 0,

.strength = 10,

.dexterity = 16,

.intellegence = 16,

.speed = 15,

.endurance = 8,

.faith = 5,

.characterClass = ROGUE

};

defaultClasses[MAGIC] = (characterType){ *// Magician class*

.hp = 25,

.mp = 20,

.strength = 5,

.dexterity = 10,

.intellegence = 16,

.speed = 16,

.endurance = 5,

.faith = 8,

.characterClass = MAGIC

};

defaultClasses[CLERIC] = (characterType){ *// Cleric class*

.hp = 20,

.mp = 20,

.strength = 5,

.dexterity = 10,

.intellegence = 8,

.speed = 16,

.endurance = 5,

.faith = 16,

.characterClass = CLERIC

};

int classChoice; *// Vars for class and name*

string name;

cout << "What is your character's name: "; *// Prompt for name*

bool valid = false;

**while**(!valid){ *// Validate input*

getline(cin, name);

**if**(!cin){

cout << "Try again. Please enter a valid name." << endl;

cin.clear();

cin.ignore(10000, '\n');

}

**else**{

valid = true;

}

}

cout << endl << "What character class do you want " << name << " to be?" << endl *// Prompt for class*

<< "1. Fighter Class" << endl

<< "2. Rogue Class" << endl

<< "3. Magician Class" << endl

<< "4. Cleric Class" << endl;

valid = false;

**while**(!valid){

cin >> classChoice; *// Getting and validating input*

**if**(!cin){

cout << "Try again. Please enter a valid number between 1 and 4." << endl;

cin.clear();

cin.ignore();

}

**else** **if**(classChoice > 0 && classChoice < 5){

character = defaultClasses[classChoice - 1]; *// Setting character to class specified by classChoice*

character.name = name; *// Applying name after so it's not overwritten*

valid = true;

}

**else**{

cout << "Try again. Please enter a number between 1 and 4." << endl;

cin.ignore();

}

}

}

*// Prompt to continue creating characters*

bool continuePrompt(){

char continueChoice;

cout << "Would you like to create another character (y or n)? " << endl;

bool valid = false;

**while**(!valid){

cin >> continueChoice; *// Validating input*

**if**(!cin){

cout << "Try again. Enter y or n." << endl;

cin.clear();

cin.ignore(100000, '\n');

}

**else** **if**(tolower(continueChoice) == 'y'){ *// Return true for 'y'*

valid = true;

**return** true;

}

**else** **if**(tolower(continueChoice) == 'n'){ *// Return false for 'n'*

valid = true;

**return** false;

}

**else**{ *// Else, try again.*

cout << "Try again. Enter y or n." << endl;

cin.ignore(100000, '\n');

}

}

**return** 0; *// This is just to clear an error, it never actually hits this*

}

# Three Use Case Screen Shots







