

National University of Computer and Emerging Sciences, Lahore Campus



Course:	Software for Mobile Devices	Course Code:	CS440
Program:	BS(Computer Science)	Semester:	Fall 2019
Out Date:	Saturday-7 th September 2019	Total Marks:	70 marks
Due Date:	Saturday-14 th September 2019	Weight:	
Section	CS-C	Page(s):	3
Assignment:	1 (Java Concepts)		

Instructions:

- Use Java Language in order to attempt this assignment
- Plagiarism of code will not be tolerated
- Create a **private** GitHub repository on your personal GitHub account with the name: **Rollnumber-JavaConcepts** ,e.g, [17L-4998-JavaConcepts]
- Upload the link to your GitHub Repository to your Google Classroom

Getting started with GitHub:

- How to configure Git on your Windows machine:
https://www.youtube.com/watch?v=J_Clau1bYco
(It is compulsory to use git bash as shown in this tutorial rather than GitHub desktop client)
While creating a new repo, make sure to make it private.
- Follow this link to add a collaborator after creating the repo:
<https://www.youtube.com/watch?v=sOciDss1EEY>
Username for collaborators: **HafsahShehzad, Mughees87**
- After creating an empty repo on GitHub, add your project to it following this link:
<https://www.youtube.com/watch?v=nHk53YnrE5k>
- For these questions, you need to develop a system that incorporates all major concepts you studied in class that primarily include: Inheritance, Polymorphism, Exception Handling, Abstract classes/methods, Interfaces, Static/Final keywords, and Data Structures
- You can ask any queries you may have on Google classroom, or on the following email:
shahzadhafsah@gmail.com

Question 1 (40 marks)

You are to develop a simple game that allows the user to select their character and interact with the opponent via interface communication.

The **game** will contain **characters** that have certain functionalities, some of which will be applicable on all characters while some which will be limited to a certain character type. All the characters of the game will be able to **jump**, **defend**, and **speak**. All characters will also maintain a **health** field that calculates the remaining power of the character after attack.

The game will contain two types of characters: **Humans** and **Aliens**. The Aliens will be the opponents to the user, and will be able to **attack**. The Humans characters will have a **name**, and will be able to **explore** and **attack**. Two types of Humans exist in the game: **Wizards** and **Warriors**. The Wizards will be able to **heal**, and attack using fireballs, while the Warriors will be able to **capture** and attack using swords.

All the functions should simply print the action of the character, e.g, Alien defend() should print: "Alien defended the attack".

The attack function should subtract the power from the character health as well. Attacks performed by aliens should subtract 5 points, attacks performed by wizards should subtract 15 points and attacks performed by warriors should subtract 20 points. Heal functions should add 10 points to the character health. Keep in mind that the health points should not fall below 0 or exceed 100. Hint: *Exception handling*.

Once the power has reached 0, the user should be notified of this with a message: "Character has been wasted".

The power should not be visible to the opponents. Hint: *Access modifiers*

The name of the Human character should only contain letters and numbers. The game should maintain a list of the characters created by the user in the game, and have should no more than 5 active characters at one point. Hint: *Exception handling*.

You have to show the interaction between at least one alien and one human character in your game. Aliens should not be able to harm aliens and humans should not be able to harm humans (Print error: "Cannot attack same character type"). Hint: *InstanceOf*

You can either use a menu to call the various functions, or initialize the characters in the main method to show interaction.

Question 2 (30 marks)

A **Fitness center** requires a system to manage the details of its **clients**. The Fitness center has four facilities: **gym, swimming pool, yoga center** and a **tennis court**.

The gym and the yoga center have trainers and are open from 10 am to 10 pm, while the yoga center and swimming pool have fixed timings from 10 am to 5 pm (Store time as military time for ease) . The tennis court and swimming pool require members to bring special gear. All the facilities are open throughout the week.

The center entertains two types of clients: **members** and **regular**. All clients are charged a fixed registration fee of \$90 when they join the gym. The regular clients pay per month, and have to pay extra in order to access each facility other than the gym (\$60 for Swimming Pool, \$50 for Yoga, \$45 for Tennis court). Make sure that the facility isn't already availed by the client.

The members pay an annual fee of \$300, along with the monthly fee that is the same as the fee paid by the regular members. This fee is decided by the center admin, i.e., the user.

All the clients have to provide their name, credit card number and age when registering at the gym. All information should not be accessed by outside classes. Hint: *Access modifiers*.

The system should store the registration date, facilities accessed and client id for each client. The system should store the total number of clients the center has of both categories, and should be able to upgrade or downgrade the client as per request.

The system should calculate how much fee and when the client has to pay their fee based on their registration date, membership type and the facilities accessed. If the client exceeds the deadline, the system should print an error message "Client has over-due fees" and remove the client from the center list after 30 days have passed after deadline.

All exceptions should be handled and appropriate data types should be used.

You are required to create a menu to show the functionalities that the system can perform.