

**Griffith College Limerick**

**COMPUTING LAB TITLE SHEET**

**Course:** B.Sc. in Computing (Level 7 and Level 8)

**Stage/Year:** II

**Module:** Object-Oriented Programming

**Semester:** I

**Lab Number:** III

**Date of Title Issue:** 18/10/23

**Assignment Deadline:** **01/11/23 22:59**

**Assignment Submission:** Submitted on Moodle

**Assignment Weighting: 10%**

## Assignment Title

|  |  |
| --- | --- |
| **Create a project called LabThree\_StudentNumber**  With StudentNumber being your student number i.e LabThree\_32323  **Add your student name and id as a comment at the top of each file.**  **Part One**  For this task you will implement the code for a basic RPG game. For each class make sure all attributes are private.  You are giving files to start with including Game, GameCharacter , Enemy, Player and the Battle interface.   * **GameCharacter** is an abstract class. Do not edit it at all * **Enemy** should extend the **GameCharacter** class and implement **Battle**   + Implement any inherited abstract methods from GameCharacter and Battle     - do not add any other methods expect inherited ones     - You can leave the behavoiur for takeDamage and heal empty for enemy, as this version doesn’t require health for enemies.   **(10 marks)**   * **Player** should extend the **GameCharacter** class and implement **Battle**   + Implement any inherited abstract methods from GameCharacter and Battle     - do not add any other methods expect inherited ones   **(10 marks)**   * Add a class called **Healer** that should extend the **GameCharacter** class   + Healers should have a cost and amount they can heal   + Add getters for both attributes   + Implement any inherited abstract methods from GameCharacter   **(10 marks)**   * Add a **Boss** class that inherits from **Enemy** * A boss has higher health than a normal enemy * Each boss is unique and can only be defeated once * Add your own unique attribute to Boss that you think fits the gameplay   **(10 marks)**   * **Game** contains the main game loop, you will need to implement the game logic here. A game is over when the player runs out of health or gets max gold. * You can edit any outputs to the console in Game but all other existing code should remain the same. * Any characters should only be accessed through the encounters list   In Game   * add instances of enemies to encounters * add one or more Bosses to encounters * add a healer to encounters   **(10 marks)**   * check if character is an enemy   + Display the enemy health and atk stat     - * + ask if the user wants to battle or run         + if they choose battle   compare atk power,  if enemies atk is higher  take away health from player  take away gold from player  increase enemies exp  increase enemies gold  continue to next encounter  if player atk is higher  add enemy exp to player  increase atk in line with exp  add enemy gold to player  if it is a Boss  use the unique attribute you gave the boss in some way  remove boss from future encounters  continue to next encounter   * + - * if attack is the same randomly choose a winner   + if they choose run then continue to next encounter   **(20 marks)**   * + - Check if character is a Healer     - Check if a user wants healing     - check if they have enough gold     - if they say yes       * increase their health       * decrease their gold   **(10 marks)**  Example Gameplay:     |  | | --- | | Note: while you **cannot** add extra methods to the instance classes you can create any extra methods you want in Game.java to complete the tasks.  **20% of marks are for comments and coding standards**   * Add a comment with your name and ID at the top of the files. * Use lowercase for variable names. * Use uppercase for class names. * Use the given names for attributes and methods. * Keep each class/interface in a separate .java file. * Keep all attributes private unless otherwise stated. * Comment any behaviours you added. * Do not create edit any existing code except for outputs and values.   **Zip the whole project folder and upload your folder as LabThree\_StudentNumber.zip** | |
|  |