

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**COURSEWORK FOR THE**

**BSC (HONS) INFORMATION TECHNOLOGY; YEAR 1 BSC (HONS) COMPUTER SCIENCE; YEAR 1**

**BSC (HONS) INFORMATION TECHNOLOGY (COMPUTER NETWORKING AND SECURITY); YEAR 1**

**BSC (HONS) SOFTWARE ENGINEERING; YEAR 1 ACADEMIC SESSION 2023; SEMESTER 2,3,4**

**PRG1203: OBJECT ORIENTED PROGRAMMING FUNDAMENTALS**

# DEADLINE: 18 DECEMBER 2023 11:59PM (Monday)

**INSTRUCTIONS TO CANDIDATES**

* This assignment will contribute 20% to your final grade.
* This is a group (maximum 5 students) assignment

**IMPORTANT**

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

Any work submitted after the deadline, or after any period of extension granted shall be marked as a Fail or awarded a zero.

**Academic Honesty Acknowledgement**

“I Lee Jing Ying, Tan Wei Tim verify that this paper contains entirely my own work. I have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, I have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. I realize the penalties *(refer student handbook undergraduate programme)* for any kind of copying or collaboration on any assignment.”

*Ying,* *Tim* (Student’s signature / Date)

# Group Number: 2

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Student ID** | **Contribution %** |
| **1** | Lee Jing Ying | 18083071 | 90 |
| **2** | Tan Wei Tim | 16035313 | 10 |
| **3** |  |  |  |
| **4** |  |  |  |
| **5** |  |  |  |

# Marking Scheme

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Reference Marks** | | **Marks** | **Remarks** |
| Design (10%)  Implement good object-oriented design in solving the problem, with high modularity, maintainability and reusability. Able to identify appropriate classes and their relationships, complete the classes with appropriate attributes and methods. Correct application of the inheritance and polymorphism concepts. The design is well presented in UML class and class relationship diagrams, and the coding is aligned to the design presented in UML. | **10** | **Excellent** |  |  |
| **7-9** | **Good** |
| **4-6** | **Average** |
| **1-3** | **Poor** |
| Coding (5%)  Fulfil all the functionalities, follow the best programming practices, such as naming convention, indenting, code structure, optimisation, with appropriate exception handling. Good user-friendliness. | **5** | **Excellent** |  |  |
| **4** | **Good** |
| **2** | **Average** |
| **1** | **Poor** |
| Add-on Feature (5%)  Implement at least one additional enhancement or feature to your program. | **5** | **Excellent** |  |  |
| **4** | **Good** |
| **2** | **Average** |
| **1** | **Poor** |
| **TOTAL** | **20** | |  |  |

**UML Diagram**

**A group of black text boxes

Description automatically generated with medium confidence**

*Figure 1. UML Diagram (full)*

**A group of black text boxes

Description automatically generated with medium confidence**

*Figure 2. UML Diagram (zoomed 1)*

**A group of black text boxes

Description automatically generated with medium confidence**

*Figure 3. UML Diagram (zoomed 2)*

**Reflection**

 i) A description of the object-oriented concepts you have applied in the assignment.

There are several concepts as follows that we have applied to this code of game.

·       Classes and Objects

·       Inheritance

·       Polymorphism

·       Encapsulation

·       Association

·       Composition

·       Exception Handling

Classes and Objects

In our code we have designed to define all the classes, each of the class will encapsulate a designated entity and concept in the system. For example, there are classes such as Pokemon, Player\_Pokemon, Game, RedeemItem, and Score. Each class serves as a blueprint for creating objects with specific attributes and behaviors in our code. We have created objects to represent the individual entity for our program. Lets say, ‘Pokemon’ objects are created to represent Pokemon, and ‘RedeemItem’ objects represent items that can be redeemed.

Inheritance

Inheritance concepts is used in the Pokémon classes such as "WaterPokemon", "GrassPokemon", "FirePokemon", "GhostPokemon", "ElectricPokemon", "PsychicPokemon" where it inherit the common attributes and methods from a base "Pokémon" class. By using the extends keyword to inherit the methods and attributes. "Player\_Pokemon" extends "Pokémon". But each of the element will have different effect towards different elements, hence we have used extend to set their own effectiveness individually.

Polymorphism

Polymorphism shown through the overriding and dynamic method. The "displayPokemonInfo" method is overridden in each Pokémon class("WaterPokemon", "GrassPokemon", "FirePokemon", "GhostPokemon", "ElectricPokemon", "PsychicPokemon"). Each method is defined by the base "Pokémon" class and is overridden by each Pokémon class to display information of a pokemon. It provides a Polymorphic behavior like the "WaterPokemon" display information of the water-type Pokémon.

Encapsulation

Public and private attributes are used in this code. Public allows interaction from outside of the class. For example: "displayPokemonInfo" method is declared as public in the base "Pokémon" class to allow external code to invoke this method.

Private attributes cannot be accessed directly from outside of the class. Therefore, there are public methods such as: "setName" and "getName" that provide controlled access to these attributes.

Association

By applying the Association relationship logic,  our Game class has associated everything in the game  with Player\_Pokemon and Score to manage gameplay and keep track of scores.

Composition

We have added composition relationships between Game and Pokemon and also Player\_Pokemon and Pokemon. In a composite relationship, the child is fully dependent on the parent class, hence when Pokemon is deleted, Game and Player\_Pokemon will also be deleted. We have designed that Game and Player\_Pokemon cannot exist alone.

Exception Handling

In order to prevent the system from crashing because of invalid inputs, we have used it in the Game class. Inside Game class, there are many inputs that require the user to type in. We cannot prevent the user from typing invalid, but we can prevent our program from crashing due to the input mismatch.

ii) A description of the add-on feature(s) you have implemented in the assignment.

Z-Move

We have set a Z-move that only when the Pokémon is capable of zMove. This can be set when the canZmove in our code is true. The battle logic is that when the Pokémon that has zMove will deal 2 times extra damage to the defending Pokemon. But this effect will only apply to the first attack, after first attack, other attack will be normal damage.

Item Redemption

When the user unsuccessfully catch the Pokémon , the system will give goldChips to the user. The user will use the goldchip to redeem items. It will first display all the items and then their description. The user can choose to save the goldChips or use the goldChip to redeem now. The item will be stored inside the system. When the user enters battlemode, it will prompt the user if they wish to use their redeemed item then only proceed to let the user to choose 2 Pokémon.

Video URL

https://youtu.be/z0\_MtOTBIyc