

**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: 22 DECEMBER 2023 11:59PM (Friday)**

**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 Chew Zhan Hong (student name).

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.”

CZH / 8/12/2023 (Student’s signature / Date)

**Group Number: 13**

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Student ID** | **Contribution %** |
| **1** | Chew Zhan Hong | 23018039 | 100% |
| **2** |  |  |  |
| **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**

텍스트, 스크린샷, 소프트웨어, 디자인이(가) 표시된 사진

자동 생성된 설명

# **REFLECTION**

Description of OOP Concepts Applied in the Program:

**To achieve a well-structured and adaptable program, this application embraces core object-oriented programming (OOP) concepts, including:**

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**Encapsulation:** Each component (Pokemon and WildPokemon) of the program is meticulously encapsulated within self-contained classes, ensuring data integrity and controlled access. This safeguards sensitive information from unintended modifications and simplifies code maintenance.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**Inheritance:** Related classes are strategically organized into hierarchies, enabling the effortless sharing of common attributes and behaviors. This promotes code reusability, reduces redundancy, and fosters a logical code structure.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**Polymorphism:** The program dynamically adapts to objects of varying types, fostering flexibility and adaptability. This is achieved through methods that can operate on diverse object types, streamlining operations and enhancing code versatility.

**The synergistic application of these OOP principles fosters a well-organized, adaptable, and maintainable codebase, ensuring the program's long-term viability and ease of evolution.**

Description of Add-On Features & Core Features in the Program:

**While I aimed to incorporate a few features that would have fulfilled the requirements such as the Catch method, switching method, Z-Move method, and the Scoreboard method. I was unfortunately unable to implement these within the current version of the program. This decision was made after careful consideration of several factors, including my group mates leaving mid-way during the assignment, having to do the entire assignment by myself with limited time, technical constraints, and restricted resources.**

**I understand that these features would have been beneficial for the overall assignment and game experience since it would have provided the complete Pokémon experience. However, I believe that prioritizing the main sequence and my own mental health was essential for delivering a playable Pokémon CLI Battle Demo based on Pokémon Ga-Ole.**

**I remain committed to exploring possibilities for incorporating this feature in future iterations, potentially through implementing a better structured system that would have been helpful in working around some of the issues. I value your feedback and will continue to assess its feasibility based on user needs and technological advancements.**

**In the meantime, I encourage you and everyone to explore the program's existing features, which offer a vanilla Battle experience exactly like the original Pokémon Games. I am confident that the current version provides a solid foundation for your immersive Pokémon battling experience and I am dedicated to the continuous improvement of this program.**

# **VIDEO URL**

[**https://www.youtube.com/watch?v=yIWCAdIFrQ0**](https://www.youtube.com/watch?v=yIWCAdIFrQ0)