

①

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION
DURATION: 3 HOURS

SUMMER SEMESTER, 2021-2022
FULL MARKS: 150

SWE 4201: Object Oriented Concepts I

Programmable calculators are not allowed. Do not write anything on the question paper.

Answer **all 6 (six)** questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.

-
1. a) Assume that you have been tasked with creating a book management system for a library. 20
 Each book has its own identifier, title, author, ISBN number, and page count. The library (CO3)
 expects the ability to add new books to its collection, remove books from its collection, and (PO3)
 search for books by title, author, or ISBN number. Additionally, there should be a feature
 that allows you to validate the ISBN number of a specific book. ISBN is available in two
 formats: 10-digit and 13-digit. An ISBN-13 number, for example, includes the following ele-
 ments: prefix element, registration group element, registrant element, publication element,
 and checksum, whereas an ISBN-10 number includes group identifier, publisher identifier,
 title identifier, and checksum. The last digit of an ISBN acts as the checksum value. Starting
 with the first digit (1-indexed), the values of the odd-positioned digits are kept the same,
 and the values of the even-positioned digits are multiplied by three; the sum of all the values
 modulo 10 is the value of the last digit (checksum). So, given a 10- or 13-digit number, you
 can use the checksum to determine whether the number is a valid ISBN.
 Write the codes of the necessary classes to imitate the scenario mentioned above. You can
 add as many classes, attributes, and methods as you need. It is recommended to use param-
 eterized constructors and getter-setter methods for properties.
- b) What is the difference between upcasting and downcasting of user-defined types? Provide 5
 examples as needed. (CO1)
(PO1)
2. Assume that you are working on a military simulation program that will simulate a war between 25
 two opposing armies. Your task is to create a program using object-oriented programming prin- (CO3)
 ciples that will allow you to simulate the movement and combat of soldiers and vehicles in the (PO3)
 warzone. The program should have the following classes:
- **Soldier**: This class should have attributes for the soldier's name, rank, health, and loca-
 tion. It should also have methods for moving the soldier to a new location and for taking
 damage.
 - **Officer**: This class should inherit from the `Soldier` class and add an attribute for the
 officer's command. It should also have a method for giving orders.
 - **Vehicle**: This class should have attributes for the vehicle's name, type, location, and
 health. It should also have methods for moving the vehicle to a new location and for taking
 damage.
 - **Tank**: This class should inherit from the `Vehicle` class and add an attribute for the tank's
 armor. It should also have a method for using a smoke screen to conceal its location. If
 the location is concealed, the method for providing tank location should throw a custom
 exception instance.

Your program should also have a unit test class to test the methods of each class to ensure they
 are working as expected. Your unit tests should include:

- A test to ensure that a soldier can move to a new location.
- A test to ensure that a soldier can take damage and that their health is reduced.
- A test to ensure that an officer can give an order.
- A test to ensure that a tank can use a smoke screen to conceal its location.

Write the codes of the necessary classes to imitate the scenario mentioned above. You can add as many classes, attributes, and methods as you need. It is recommended to use parameterized constructors and getter-setter methods for properties.

3. a) Assume that you are working on a project to create a logger class that will be used throughout your application to log messages to a text file at different levels of severity. You want to ensure that there is only one logger instance in your application and all the classes in your application can access the same instance of the logger. To achieve this, you decide to implement the logger class using the singleton pattern. The logger should have the following features: 14
(CO3)
(PO3)

- It should be able to log messages to a file, including the message level, timestamp, and source of the message.
- It should support different message levels, such as DEBUG, INFO, WARNING, and ERROR.
- It should provide a global point of access to the logger instance, so that any class in your application can log messages to the same file.

Write a class for the logger that implements the singleton pattern, and demonstrate how it can be used in your application. Also, write a brief explanation of how the singleton pattern is useful in this scenario.

- b) A software company has developed an accounting program that has multiple modules, each of which can be loaded into the main program at run time if that module is installed. The company wants to apply updates to specific modules without affecting other parts of the program. Which approach would you recommend to the company, using DLLs or not using DLLs? Explain your reasoning and provide the advantages and disadvantages of each approach. 7
(CO2)
(PO2)
- c) How can you pass an instance of a primitive data type to a method by reference using C#? Explain with necessary examples. 4
(CO1)
(PO1)

4. Assume that you are tasked to design a system that simulates a university. The university has several departments, including Computer Science, Mathematics, and Physics. Each department has many courses, and each course has many students. Students can take multiple courses in different departments. The system should keep track of each student's information, including their name, ID number, and the courses they are enrolled in. Each department has a head who is responsible for managing the department's courses and instructors. Each course has a unique course ID, a course name, and a list of instructors who teach that course. Instructors have a name, an ID number, and a list of courses they teach. The university also has a library, which has a collection of books. Each book has a unique ISBN number, a title, and an author. Students can borrow books from the library and return them when they are done.

- a) Draw the UML class diagram along with cardinality information for the scenario in Question 4. 10
(CO1)
(PO1)

b) Answer the following questions:

- i. Identify the classes in the university system and the relationships between them using inheritance, association, aggregation, and composition.
- ii. Write a code snippet that demonstrates the association relationship between the Student and Course classes.
- iii. Explain how the aggregation relationship is implemented between the Department and Course classes.
- iv. Suppose that a student wants to borrow a book from the library. Which relationship between classes is involved in this scenario, and how is it implemented?
- v. Describe the composition relationship between the Department and Head classes. What happens when the Department object is destroyed?

5. a) A software company is developing a new project that will handle data processing for various types of clients. The project involves creating different classes for different data types, including strings, integers, and floating-point numbers. However, the company is concerned about code redundancy and wants to ensure that the project can be easily maintained and scaled in the future. How can you use Generics in the design of the data processing project to reduce code redundancy and ensure the project can be easily maintained and scaled in the future? Provide necessary examples. 10
(CO1)
(PO1)

b) ESILES Digital Platforms is a renowned software firm in Eriador, Middle-Earth. They are developing an employee management system for themselves. So far they have come up with the following system architecture. There is a base Employee class that will have two child classes, BusinessEmployee, and DeveloperEmployee. There is also another type of employee that handles client-side operations as well as development issues. So, they have created another class named, ProductEngineer. The ProductEngineer class inherits both from BusinessEmployee and DeveloperEmployee classes. The class diagrams for these four classes are given in Figure 1. The parent Employee class is an abstract class 5+10
(CO1)
(PO1)

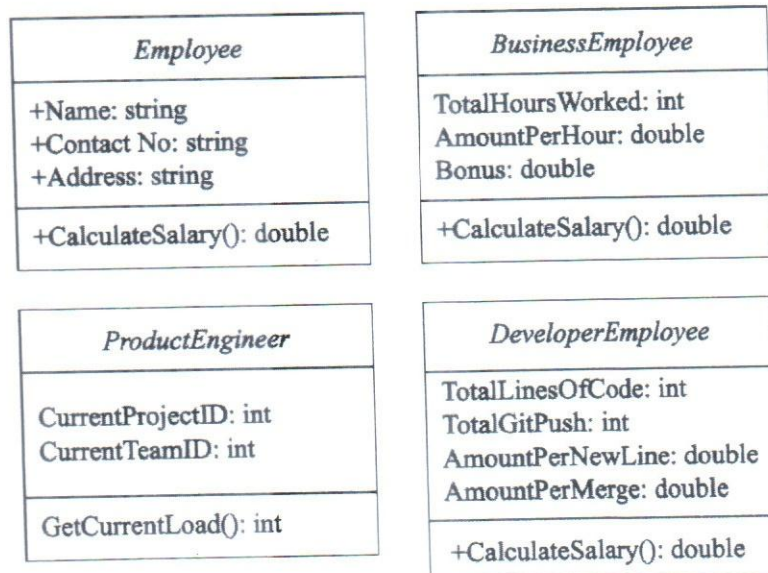


Figure 1: Classes associated with the Employee Management System of Question 5.b)

since it contains an abstract method, CalculateSalary(). BusinessEmployee and DeveloperEmployee classes override this method. Answer the following questions:

- i. It is not possible to use C# to construct this same class architecture. Provide the relevant explanations for why this type of implementation is prohibited.

- ii. Write necessary codes to implement the aforementioned scenario considering the four types of employees adhering to the following constraints.
- Multiple inheritance is permitted only when there is at most one base class and the rest are interfaces.
 - For the class methods (`CalculateSalary()` and `GetCurrentLoad()`), you can use any implementation that makes use of the existing class attributes.
6. Consider a game named "Guess It Right". In this game, the user is given a random number and asked to estimate whether the following number is greater or lesser than the current number. As long as the player's guess is correct, the game continues. The player receives one point for each correct estimate. The numbers that appear are always even and between 10 and 100. The game also saves the player's name, last played score, and all-time highest score. If someone beats the highest score, the prior scorer's name is removed and replaced with the new scorer's name. A player's name cannot exceed 20 characters and must begin with an uppercase alphabet and end with a lowercase alphabet. The game also maintains track of how many times it has been played since it first started. After the game begins, it reads prior records and all information from a file called "data.txt," and when the game is finished, it rewrites the new information into the same file.

In order to play the game, the following functionality must be implemented:

- Both parameterized and non-parameterized constructors should be available. The highest score as well as the name of the scorer will be passed to the parameterized constructor.
- There should be a private method with the method signature shown below.

```
1 private bool VerifyName (string name);
```

This method will return true if the player's name is valid and false otherwise.

- There should be two private methods that will contain the following method signatures.

```
1 private bool UpdateLastScore (string name, int score);  
2 private bool UpdateHighScore (string name, int score);
```

These methods will be responsible for updating the score information in the text file.

- There should be a public method, `Play()` that will coordinate all the functions and simulate the game.

To imitate the game mentioned above, create a class called `GuessGame` and include the necessary and appropriate member variables and functions. Inclusion of any attributes or member methods to simulate the game is permitted.