ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION **SUMMER SEMESTER, 2021-2022**

DURATION: 1 HOUR 30 MINUTES **FULL MARKS: 75**

SWE 4201: Object Oriented Concepts I

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

Answer **all 3 (three)** questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.

1. a) Write down the output for the program given in Code Snippet 1. 10 (CO2) (PO2)

```
1  using System;
2  using System.Collections.Generic;
3  namespace OOCI {
4      public class Shape {
5          public int X { get; private set; }
6          public int Y { get; private set; }
7          public virtual void Draw() {
8              Console.WriteLine("Performing base class drawing tasks");
9          }
10     }
11     public class Circle : Shape {
12         public override void Draw() {
13             Console.WriteLine("Drawing a circle");
14             base.Draw();
15         }
16     }
17     public class Rectangle : Shape {
18         public new void Draw() {
19             Console.WriteLine("Drawing a rectangle");
20         }
21     }
22     public class Square : Shape {
23     }
24     class Program {
25         static void Main(string[] args) {
26             var shapes = new List<Shape> {
27                 new Rectangle(),
28                 new Circle(),
29                 new Square()
30             };
31             foreach (var shape in shapes) {
32                 shape.Draw();
33             }
34             Shape baseObj = new Circle();
35             baseObj.Draw();
36             Circle derivedObj = (Circle)baseObj;
37             derivedObj.Draw();
38         }
39     }
40 }
```

Code Snippet 1: C# code snippet for Question 1.a)

b) Explain the architecture of the.NET framework and the responsibilities of the submodules. Use figures where necessary. 7 (CO1) (PO1)

c) Explain the following terms with necessary examples. 2×2
   i. copy constructor (CO1)
   ii. private protected (PO1)

d) "Auto-implemented properties cannot have any logic in their accessors. An auto-implemented property can however have different access modifiers for its accessors." - Explain the statement with necessary examples. 4 (CO1) (PO1)

2. Department of CSE, IUT is going to organize the 11th ICT Fest. The ICT Fest has two events: a Programming Contest and a Gaming Tournament. There are two types of video game competitions: Need for Speed (NFS) and Defense of The Ancients (DOTA). NFS requires players to compete as individuals, while programming competitions and games like DOTA demand teams of three. 20 (CO3) (PO3)

   In order to track all the participants who have registered to attend this festival, an application is required. The system will store each participant's unique integer *ID*, their *name*, *address*, *institution*, and *t-shirt size*. Each registered team in the programming contest will have the following information stored in the system: *team ID, team name, participants (three), institute, total obtained points, preferred programming language*, and *allocated room number*. Each team in the programming competition will have access to four methods: *SubmitSolution*, *PayFees*, and *GetTShirt*. The *SubmitSolution* method will generate a random integer between 0 and 100. The generated number will be added to the respective team's total points. Since the fee for the Programming contest is BDT 3000, the *PayFees* method will return the string "Team {team_name} has paid BDT 3000". Here, {team_name} will be replaced with the name of the corresponding team. The *GetTShirt* method will print a team's participants' names as well as their T-Shirt sizes.
   For gaming event teams, *team ID* and *team name* will be stored in the system along with a boolean attribute, *isEliminated*. Each team in the gaming event will have access to three methods: *GetTShirt*, *PayFees*, and *Eliminate*. *Eliminate* will set the isEliminated property to True. NFS costs BDT 500 per participant, while DOTA costs BDT 1200 per team.

   For the ICT Fest, there will also be an *account* class that will keep track of the current balance. The amount paid by a team or participant will be added to the current balance. The account class will also include a method for displaying team or participant IDs that are yet to pay their fees. The *account* class will provide these features without the need to create an instance.

   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. Tristan is a Cornish knight, whereas Morholt originates from Ireland. The destiny of princess Iseult hangs in the balance as these two knights engage in a sword duel. If Tristan prevails, Iseult will be compelled to move to Cornwall, whereas if Morholt does so, she will be allowed to remain in Ireland. The archer Rivalin is a good friend of Tristan's, yet he serves in the Irish Army. So, he does not choose a side and instead launches arrows at the sword combat without thinking. It may strike Tristan, Morholt, or something completely unrelated. The victor shall be the knight who is the last one standing. **You have to write the codes for the classes needed to model the entities described below.** 30 (CO3) (PO3)

   A utility class with two methods is needed for the scenario. One will produce a random integer that is either 0 or 1. The second one will produce a random integer between 0 and 2 (inclusive). The following code snippet generates a random decimal between 0 and 1:

```
1 static void PrintRandom() {
2     Random rand  = new Random();
3     double num = rand.NextDouble();
4     Console.WriteLine(num); // Sample output: 0.428724402761424
5 }
```

Code Snippet 2: A C# method to generate a random decimal between 0 and 1 for Question 3.

Some characteristics are shared by all soldiers, whether they are knights or archers. *ID, name, and health*. The range of possible values for health is 0-100. A soldier will expire when his health bar reaches 0. The archer will use arrows and the knight will use a sword. There is a wide variety of swords available, each made from a unique metal with its own unique properties and damaging potential. For example, if a sword causes X points of damage, it will reduce a soldier's health by X points if it hits him. This context requires X to be a positive integer between 5 and 25. If the sword's current damage is more than 5, it will be reduced by 5 after every attack. However, each archer will be armed with ten arrows and each will do the same amount of damage, 20. Any time an archer fires an arrow, he loses one of his arrows.

When a knight is hit by an arrow or a sword, his health is reduced in accordance with the weapon's damage value. In each assault, a knight will use his sword to strike his opponent (another knight) or a random item. The utility class will be used by the knight's *attack* method to generate the random number. If it is 0, the knight will strike a random item with his sword; if it is 1, he will strike his opponent. Each archer will have a similar *attack* method. If the randomly generated number is 0, the discharged arrow will strike a random item. If it is 1, it will hit Tristan; if it is 2, it will hit Morholt.

Requirements:

- Your solution should incorporate the concept of inheritance and polymorphism.
- There should be at least one sealed class.
- There should be at least one abstract class and method.
- There should be at least one static class.
- Parameterized constructors and getters-setters for attributes are required for all non-static classes.