B.Sc. Engg. SWE 2nd Semester Libery 6 November 2019

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION **SUMMER SEMESTER, 2018-2019**

DURATION: 3 Hours **FULL MARKS: 150**

SWE 4201: Object Oriented Concepts I

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

There are **8 (eight)** questions. Answer any **6 (six)** of them.

Figures in the right margin indicate marks.

1. a) Define class and object with example. Explain the relationship between class and object. 5+5

   Identify 5 objects from the paragraph below. For each of the objects, mention which type the object belong to. You may identify multiple objects of same type. Note that, the type of object may or may not be mentioned in the paragraph.

   "The Treaty of Hudaybiyyah took place during the time of the prophet Muhammad (S). It was a crutial treaty between the Muslims and the Qurayshi tribe of Makkah in the 6 Hijri year. Many of the sahabas, including Umar (R) initially objected to it because it seemed to be a humiliation and defeat to the Muslims. However, Quran mentioned this as a victory."

   b) Consider the following items:
   • Book • Humayun Ahmed • Facebook • Publisher • Today • Bag • Social Media • Person • Suitcase • Author

   i. Which items above most naturally belong to class category and which ones most naturally belong to object category? 5

   ii. Give at least two examples of Is–a relationship among classes in these items. 2

   iii. Give at least two examples of Has–a relationship among classes in the items above. 2

   c) How does method overloading differ from method overriding? Give example. 6

2. a) Define the terms *coupling* and *cohesion* with example in the context of object oriented concept. Within your discussion, explain how *coupling* and *cohesion* can lead to either good or bad software design. 11

   b) What is *abstraction*? Give two real life examples of *abstraction*. Give example in an object oriented programming language of your choice. 8

   c) What are *abstract class* and *interface*? Give differences between an interface and an abstract class. 6

3. a) Define *class variable* and *instance variable*. Provide a code fragment that implements these concepts in an object oriented programming language of your choice. 8

   b) What are access specifiers? What is the use of access specifiers? 4

   c) Explain why it is considered good practice to limit the scope of fields and methods in object oriented programming. 3

   d) The following table is supposed to define the access level to a field of a class permitted by each modifier in java. Fill in the blanks of the following table with "Yes" if the field is accessible from the respective class or "No" if it is not. 10

	Private	No Modifier	Protected	Public
Same class				
Subclass in same package				
Subclass in different package				
Non subclass in same package				
Non subclass in different package				

4. a) Define aggregation and composition with example? How does composition differ from aggregation? Why do we need composition-aggregation? 10

   b) Explain the followings with logic: 6+4

      i. When do we use composition over inheritance and vice versa (inheritance over composition)?

      ii. Does composition increase coupling?

   c) Using an object oriented language with which you are familiar, give an example of delegation. 5

5. a) Explain the followings with diagram: 3×3

      i. Single inheritance

      ii. Multiple inheritance

      iii. Multilevel inheritance

   b) Organize the following classes into inheritance hierarchies and where appropriate create new classes: `Animal, 3DShape, Zebra, Employee, Cube, Shape, SalariedEmployee, Circle, PetAnimal, Consultant(Part Time Employee), Bird, 2DShape, Cat, Manager, Fish, Triangle, HourlyEmployee.` 16

6. a) How do we represent the followings in UML class diagram?

      i. Visibility of class members as private, protected, public and default 4

      ii. Aggregation and Composition 4

      iii. Multiplicity of association (one to many and many to many relationships) 4

      iv. Inheritance from class, abstract class and inheritance 6

   b) Write down the code (in an object oriented programming language) for the following UML class diagram. 7

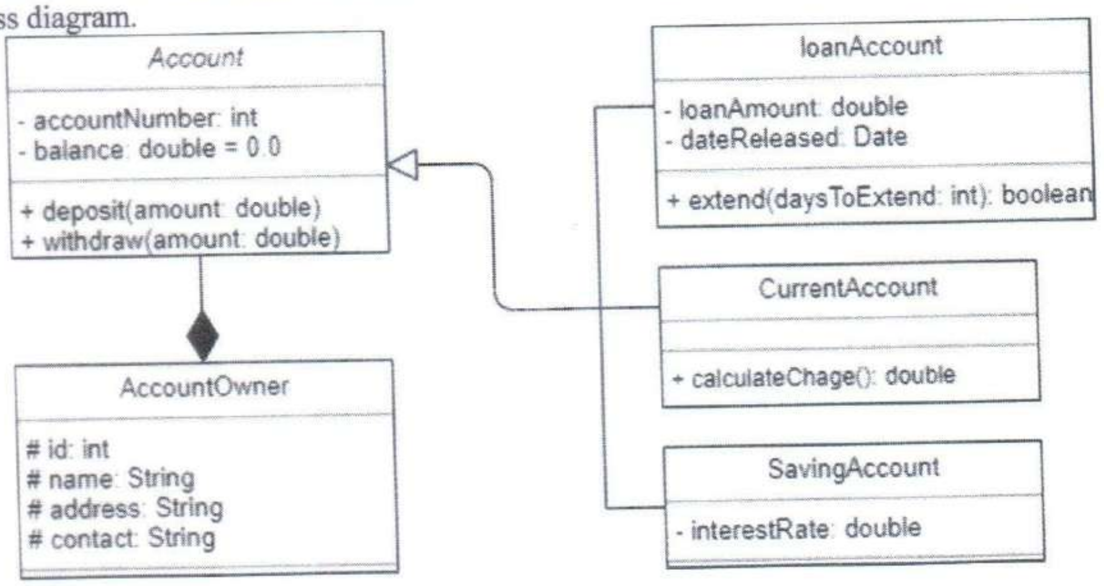

Figure 1: UML class diagram for question 6.b.

7. In the game "homely", the player can arrange a house with his own design. The house has a predefined size (area). The house can have different types of rooms like bedroom, kitchen, toilet, veranda. The total area of the rooms must be equal to the size of the house. Also, you can paint any room with any color. In each room, there can be any number of windows and doors. The windows and doors can be opened and closed, and also can be locked. Player can place any furniture in any place of the room and can move them later. The furniture is made of different materials. There can be two types of furniture: furniture without lock and with lock. 25

   Now design the scenario from your knowledge of object oriented concepts. You can make the design in a UML class diagram or in any object oriented programming language. If you use UML class diagram, give proper detail of each class, its properties and methods, and relations.

8. Examine listing 1 which contains a section of java code that has several errors. Locate 5 errors in the code and explain why you believe there is an error at that location. Write the correct code where there is error. 25

Note: When you are thinking about one line. Consider that rest of the code has no errors.

```
 1.  class Phone {
 2.     private String phoneNumber;
 3.
 4.     public void call(String toNumber) {
 5.        System.out.println("Calling from " + phoneNumber + " to " + toNumber);
 6.     }
 7.  }
 8.
 9.  class MobilePhone extends Phone {
10.     public void sendSms(String toNumber, String text) {
11.        System.out.println("Texting from " + phoneNumber + " to " + toNumber);
12.        System.out.println(text);
13.     }
14.
15.     @Override
16.     public void call(String toNumber, boolean isVideoCall) {
17.        if (isVideoCall)
18.          System.out.println("Video Calling from " + phoneNumber + " to " +
                   toNumber);
19.          else
20.             super.call(toNumber);
21.     }
22.  }
23.
24.  abstract class Radio {
25.     public void play(String chanel) {
26.        System.out.println("Playing " + chanel);
27.     }
28.  }
29.
30.  class AndoidPhone extends MobilePhone, Radio {
31.     public void connectToInternet(String wifiName) {
32.        System.out.println("Connecting to " + wifiName);
33.     }
34.  }
35.
36.  class Main {
37.     public static void main(String[] args) {
38.        Radio r = new Radio();
39.        r.play("12.8");
40.
41.        MobilePhone m = new Phone();
42.        m.sendSms("01xxx xxxxxx", "Salam!");
43.     }
44.  }
```

Listing 1: Erroneous Java code for question 8.