UNIVERSITY OF RUHUNA
BACHELOR OF SCIENCE (GENERAL) DEGREE
LEVEL II (SEMESTER I) EXAMINATION – AUGUST 2017

COM212β – Object Oriented System Development **Duration: 2 hours**

Answer **four** questions **only**

1)

a) What is a **Driver class**?

b) Name the **two (2)** categories of attributes used in classes. Describe the key difference between them.

c)

i. Briefly explain the role of **Access Modifiers**.

ii. Describe the difference between **public** and **protected** modifiers in **Java Programming** with respect to the accessibility aspect.

iii. Consider the **Person Class** and **PersonDrive Class** given below.

Preson Class

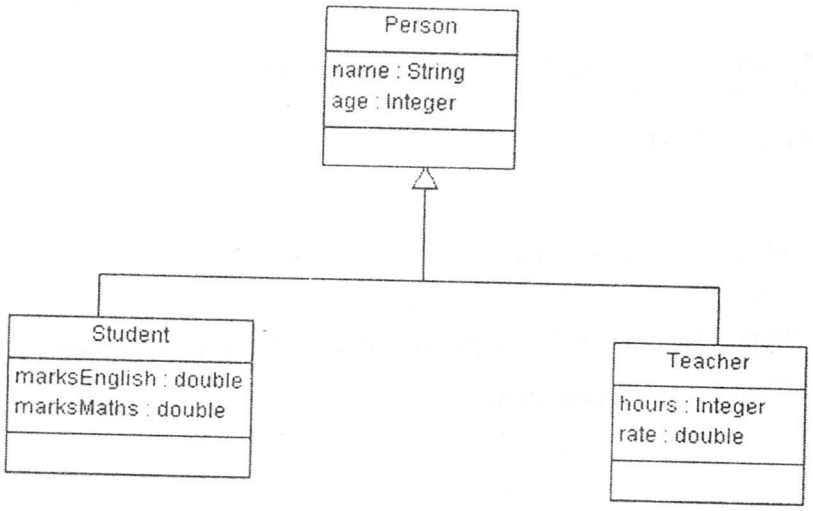
```
class Person
{
        public String name;
        private int age;
}
```

PresonDrive Class

```
class PersonDrive
{
        pubic static void main(String[] args)
        {
                Person bob=new Person();
                bob.name="Bob"; ← (A)
                System.out.println(bob.age);← (B)
        }
}
```

State whether each of the statements **(A)** and **(B)** in the PersonDrive class are valid or not. Explain your answers.

d) Consider the partial class diagram given below.



i. What is the **Object Oriented Design** concept that is used to link the **Student,** and **Teacher** classes with the **Person** class? State an advantage of using the concept you mentioned.

ii. State how the concept you mentioned in **(i)** above is facilitated in **Java Programming**.

iii. Write **Java code** segment that implements the design given in the above class diagram (no need to consider access modifiers).

2)

a) List four (04) best practices of the **Rational Unified Process (RUP).**

b) Briefly describe the following **Object Oriented Design Principles**.

i. Abstraction

ii. Polymorphism

c)

i. What is the role of a **constructor** in **Java programming**?

ii. What is meant by the **default constructor** of a class?

iii. Consider the code segment given below.

```
class Course
{
        String course_id;
        String name;
        double credits;
}
```

Write **parameterized constructor** using **Java** to assign values to each attribute in the above class.

d)

i. State the key difference of **abstract classes** with respect to non-abstract classes.

ii. Briefly explain what is meant by **Method Overriding**.

iii. Write a **Java** code segment to create an abstract class called **AbsMessage** that has an abstract method called **displayAlert()**, which returns a **String** value.

iv. Create a class called **MessageBuilder** that inherits from the abstract class you created in **(iii)** above. Override the method **displayAlert()** to return the message: *"Message Builder Alert"*

3)

a) Briefly describe **two (2)** types of **UML** diagrams used in the analysis phase of the Rational Unified Process.

b) Write appropriate sequence of activities that should be followed in developing a class diagram.

c)

i. Briefly explain the **Composition** relationship between classes. Give a suitable example using a class diagram.

ii. Consider the following description

A research project is handled by at least one department in the university, but maximum two departments can involve in handling a particular research project. Also, a department can have zero or more research projects to handle.

Draw a UML class diagram depicting above scenario.

d) The following facts are collected to design a software to handle the activities in a vehicle manufacturing firm. Draw a UML class diagram based on the given details.

- A vehicle is a collection of vehicle parts.
- Vehicle parts are manufactured separately and are used to assemble a vehicle.
- Vehicle parts of a vehicle are replaceable.
- Cars and Lorries are the vehicles manufactured in the firm.
- An assembled vehicle is checked by an employee.
- An Employee can check one or more vehicles.

4)

a) What is the key objective of **Use Case Analysis**?

b) List **four (4)** benefits of **Use Case analysis** in Software Development.

c)

i. Briefly explain the usage of **extends** relationship between use cases in use case diagrams.

ii. State an advantage of **include** relationship between use cases.

d) Consider the following scenario regarding an online railway reservation system. Develop a **Use Case diagram** based on the given details.

Online railway reservation system provides several facilities to customers. The System allows customers to search the availability of trains. After searching, customer can make a reservation request. Before making the reservation request, the customer should be authenticated by the system.

A clerk in a station can log into the system and can reserve seats based on the reservation request made by a customer. She can reject any reservation request if seats are not available or the customer has not submitted required details about the journey. An invoice is sent to the customer when a reservation is done. Once the customer makes the payment, the clerk processes the payment and the reservation is confirmed.

5)

a) What do **UML Sequence Diagrams** model?

b)

i. Briefly describe what is represented by **Lifelines** and **Activations** in Sequence Diagrams.

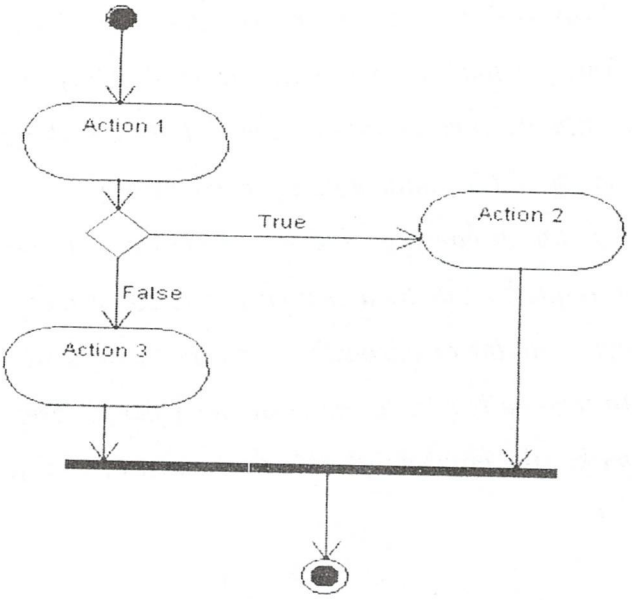
ii. Explain the difference between **Synchronous Messages** and **Asynchronous Messages** with respect to Sequence Diagrams.

c) Consider the following scenario related to obtain a digital certificate for a public key.

In this process, first the web user has to generate a public key. After that, it is sent to a certification authority. Once the public key is received, the certification authority checks the identity of the web user. If the user is authentic, a certificate is created to the web user's public key. At the same time, certification authority updates their logs. Finally, the generated public key certificate is sent to the web user. However, if the user is not authentic, a failure notification is sent to the user.

Draw a UML Activity diagram for the above scenario showing **swim lanes** (partitions).

d) Consider the following Activity diagram.



i. Explain the problem associated with above activity diagram.

ii. Suggest a solution to solve the problem you explained (i) above.