

# **University of Ruhuna - Faculty of Technology**

**Bachelor of Information & Communication Technology Degree**

**Level 2 (Semester 1) Examination**

**November 2019**

**Course Unit: IC T2123, Object Oriented Development**

**(Theory)**

**Time Allowed: 02 hours**

---

This question paper contains **eight (08) pages** including this instruction page

## **IMPORTANT INSTRUCTIONS:**

1. The medium of this examination is **English**.
2. This is a **Closed Book** examination.
3. This Examination consists of four (04) questions that are given equal marks.
4. You must answer **all four (04) questions** in this examination.

1. a. i. List down five (05) key features of Java.
- ii. Briefly describe what is known as "Write once, Run everywhere" concept related to Java?

b. Write down the answers for following question by considering the elements and their order in a Java Class.

Element	Example Code Segment	Required (Yes/No)	Where it's appearing in the Class (Select from the below given List A)
Comments			
Package declaration			
Import statements			
Class declaration			
Method declaration			

**List A**

Anywhere inside the class scope,  
Immediately after the package,  
Immediately after the import

Anywhere,  
First line in the file,

- c. i. List down two (02) key differences between "Java Primitive Data Types" and "Java Reference Data Types".
- ii. By using two (02) key points briefly describe the difference between "Java Object" and "Java Object Reference".
- d. Investigate the following java code and answer the questions given below.

```
public class ForLoopDemo {
    public static void main(String[] args){
        int age = 10;
        String names[] = { "Nimal", "Kamal" };

        for ( int i = 0, age = 10 ; i < names.length ; i++)
        {
            String name = names[i];
            System.out.println(name + " , " + age);
        }
    }
}
```

- i. What will be the output when you compile and run the above Java program. Explain the reason/for your answer.

- ii. Write down the Java code changes which needs to be done in above class in order to get following output.

```
Nimal , 10
Kamal , 15
```

- iii. Replace the for loop with a for-each loop to obtain the same output in part (d) (ii) section.

- e. Investigate the following Java code and answer the questions given below.

```
public class Person {
    String name = "Nimal";
    int age;

    public (String name){
        name = this.name;
    }

    public static void main(String[] args){
        Person kamal = new Person( "Kamal" );
        System.out.println( kamal.name );
        System.out.println( kamal.age );
    }
}
```

- i. What will be the output of the program. Explain the reason/s for your answer.
- ii. Write down the required changes in the above given code to obtain following output by using constructor chaining concept in Java.

```
Kamal
20
```

2. a. Using following scenario, answer the questions given below.

Assume that there is a high secured front door in your faculty main building. Since you are an ICT student, Head of the Institution ask you to write a Java program to open the door based on following requirements.

Case 1: Door can be opened using a Finger Print of an employee

Case 2: Door can be opened using an Iris of an employee

Case 3: Door can be opened using a Key Code

- i. What is the OOP concept that you can use to implement the Java code to satisfy the requirements of the above given scenario?



- ii. Write down the **method signatures** of the "openDoor" java method which satisfy the below listed conditions for the **three (03) cases** of the above given scenario.

- method is not returning anything
- method is publicly accessible from anywhere

- iii. Briefly describe the **Dynamic Method Dispatch** concept in Java.

- iv. **Complete** the main method of the following java code to demonstrate the **Dynamic Method Dispatch** concept in Java.

Write down the expected **output** as well.

```
public class Car{
    void start(){
        System.out.println("Car is starting");
    }
}
public class Toyota extends Car{
    void start (){
        System.out.println("Toyota car is starting");
    }

    public static void main(String args[]){
        //Your Code
    }
}
```

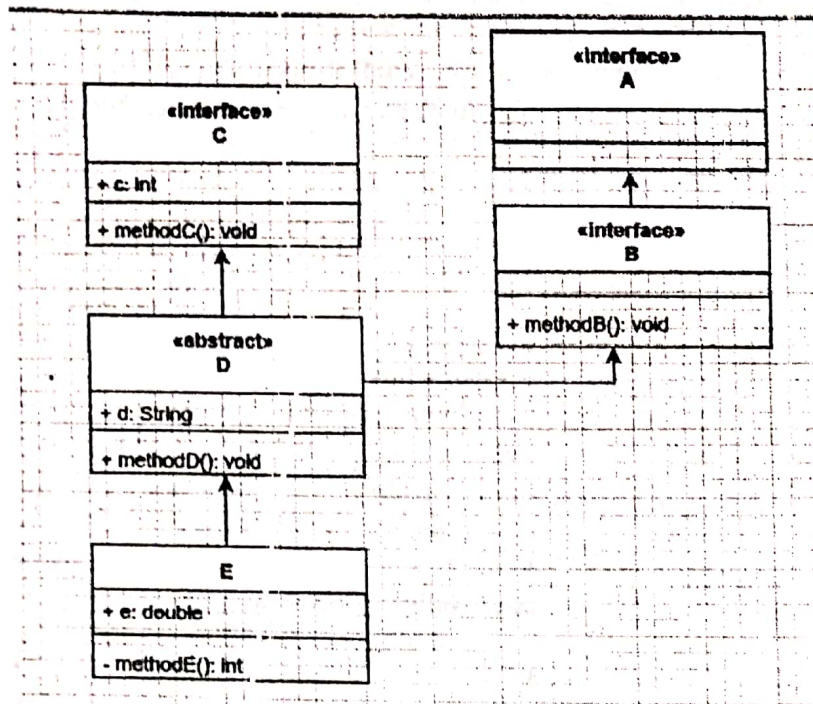
- b. By considering the Java code in following "Eligibility" Java class answer the questions given below.

```
public class Eligibility{
    private boolean eligible;

    public boolean isEligible (){
        return eligible;
    }
    public void setEligible (boolean eligible){
        this. eligible = eligible;
    }
}
```

- i. What is the **OOP concept** used in the above "Eligibility" Java class?
- ii. Identify and write down the **accessors** and **mutators** in the above Java class.
- iii. Identify the changes and **Rewrite** the **Eligibility** class as an **Immutable** class.

- c. Consider the following class diagram and answer the questions.



- i. Write Java code segments for B, D and E in the above class diagram.
- ii. Write down the special name given to the interface A.  
Write down why we are using this special category of interfaces in Java inheritance.
- iii. Write down the method signature for "methodD" if it's forbidden to override by a child class.

3. a.
  - i. Briefly describe following exceptions and errors in Java.
    - NullPointerException
    - FileNotFoundException
    - StackOverflowError
  - ii. By using two (02) key points briefly describe the difference between "throw" and "throws" in java exception handling.



- iii. Consider the below given code segment.

```
public class PrintNumbersDemo {  
    public static void main(String[] args) {  
        PrintNumbers pn = new PrintNumbers();  
        pn.start();  
    }  
}
```

```
public class PrintNumbers {  
    public void start() {  
        System.out.print("1");  
        try{  
            divide();    //Line A  
            print();    //Line B  
        }  
        catch(NullPointerException e){  
            System.out.print("2");  
        }  
        finally{  
            System.out.print("3");  
        }  
        System.out.print("4");  
    }  
  
    public void divide() throws ArithmeticException{  
        System.out.print("5");  
        int x = 5/0;  
        System.out.print("6");  
    }  
  
    public void print() throws NullPointerException{  
        System.out.print("7");  
        String a = null;  
        a.toString();  
        System.out.print("8");  
    }  
}
```

- Write down the output when you compile and run the above Java program. Write down the reasons for your answer.
- Write down the output when you interchange Line A with Line B and compile and run the above Java program. Write down the reasons for your answer.

- b.
- i. Write down the five (05) major states of a life cycle of a thread.
  - ii. Briefly describe "Dead Lock" in Java multithreading.
  - iii. Consider the java code given in the "Counter" Java class.

```
public class Counter{
    public void printCounter(){
        PRINT_LOOP : for( int i =0 ; i < 3 ; i++){
            System.out.println( i );
        }
    }
}
```

Write down the code segments required to make PRINT\_LOOP (for loop) synchronized by using following methods.

- by using Synchronized Methods
- by using Synchronized Statements/Blocks

- 4.
- a.
    - i. Write down three (03) methods used in Java JDBC Statement Interface to execute SQL statements in Java.
    - ii. Consider the employee table given below in a MySQL Database.

Table : employee

id	name	age
1	Niaml Perera	25
2	Kamani	20

Write down the code segment to update employee details of "Kamani" using PreparedStatement as follows.

Name: Kamani Perera  
Age: 23

(Note : you have to write only the code from preparing statement to execution of the statement)



b. Use following description to answer the below given question.

Assume that you are recruited as a software engineer for a Software firm and your task is to develop an employee management system for the company. MySQL Back End and Java Front End has already developed by another team and your first task is to develop the connection between the Back End and Front End. As informed by the management of the company they have license for **only one connection** for their database. So, you have to utilize that **Single Connection** object across the whole application.

Connection details are as follows.

Host Name: localhost	IP-Address: 192.168.1.24
Database Name: techno	Port No: 3306
User Name: admin	Password: 1dm9n

Connection class name as per the design: **DBConnectionManager**

Using your knowledge in **Design Patterns**, Write down the Java code of the "DBConnectionManager" class.