

**University of Ruhuna - Faculty of Technology**  
**Bachelor of Information & Communication Technology Honours Degree**  
**Level 2 (Semester I) Semester Examination, June/July 2023**  
**Academic year 2022/2023**

**Course Unit: ICT 2122 - Object Oriented Programming**  
**(Written)**

**Duration: 02 Hours**

.....

This question paper contains six (06) 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.

**An object-oriented system is a set of interacting objects organized into classes.**

i. Briefly describe the following concepts used in Object Oriented Programming.

- Instance
- Instantiation

[10 marks]

ii. Consider a public java class named *Vehicle* which is written in a file named *Vehicle.java*. Fill out the following table by considering the *Elements of Vehicle class* and their order in the file using *List A*.

Element	Sample Code Segment (Select from <i>Vehicle elements</i> )	Required Always (Yes/No)	Where it's appearing in the file (Select from the below given <i>List A</i> )
Package declaration			
Import statements			
Class declaration			
Method declaration			
Field declaration			

**Elements of Vehicle class**

```
public Class Vehicle { },      private String name; ,
import java.util.io; ,      package com.ictec.veh;      public void run(){ }
```

**List A**

Anywhere inside the class scope,      Anywhere,      First line in the file,  
Immediately after the package,      Immediately after the import

[15 marks]

b.

Consider the four (04) *Access Modifiers* which are used in Java.

Fill out the following table by selecting all the letters which are applied from *List B*.

Member is	Can accessed by (Select all the letters (A,B,C,D) which are applied from <i>List B</i> )
<b>public</b>	
<b>protected</b>	
<b>default</b>	
<b>private</b>	

**List B**

- A - A member in the same class
- B - A Member in another class in same package
- C - A Member in a superclass in a different package
- D - A Member in a non-super class in a different package

[15 marks]

c. *Control statements are the statements that change the flow of execution of statements.*

i. What is the output of using the following *for loop* inside a Java program?

```
for( ; ; ) {  
    System.out.println("Hello World...!!!");  
}
```

[05 marks]

ii. What will be the output when you compile and run the above *MyArrayDemo* class? Give your reasons.

```
public class MyArrayDemo {  
    public static void main(String args[]) {  
        int sum = 0, x = 5, y = 0;  
  
        for( ; x > y ; x--, y++) {  
            sum += (x+y);  
        }  
        System.out.println(sum);  
    }  
}
```

[15 marks]

d. *A constructor in Java is a special method that is used to initialize objects.*

i. List down *two (02)* unique features of the default constructor used in a Java class.

[10 marks]

ii. Consider the given *MyDemo* and *MyClass* classes below.

```
public class MyDemo {  
    public static void main(String args[]) {  
        MyClass ms = new MyClass(); // → Line 3  
    }  
}
```

```
public class MyClass {  
    int a = 3;  
  
    {  
        System.out.print(a);  
    }  
  
    static {  
        System.out.print(4);  
    }  
  
    MyClass() {  
        System.out.print(a+2);  
    }  
  
    MyClass(int y) {  
        this();  
        this.a = y;  
        System.out.print(a+3);  
    }  
}
```

What will be the output when you compile and run the above *MyDemo* class? Give your reasons.

[15 marks]

- iii. What will be the output when you compile and run the above *MyDemo* class by replacing the code *Line 3* with the following code line. Give your reasons.  
*new MyClass(4);*

[15 marks]

2.

a. **Inheritance is one of the major pillars in Java OOP.**

- i. By using an example briefly describe the concept of inheritance.

[10 marks]

- ii. Consider the given *Animal* and *Cat* classes below.

<pre>public class Animal {     //Rest of the code }</pre>	<pre>public class Cat extends Animal {     //Rest of the code }</pre>
---	---

A. Identify and write down the following things.

- Sub Class
- Indirect Super Class

B. By giving your reasons state whether the following object creations are valid in Java.

- `Animal a = new Animal();`
- `Animal a = new Cat();`
- `Cat c = new Animal();`
- `Object o = new Animal();`

[30 marks]

b. Consider the given *MyData* class below.

```
public class MyData {
    public void printMyData ( String a, String b ){
        System.out.println ( a + " , " + b );
    }
    public void printMyData ( double x, double y ){
        System.out.println( x + y );
    }
}
```

- i. What is the OOP concept used in the above *MyData* class?

[05 marks]

- ii. Write down the output when you invoke the *printMyData* method with the values given below. Give your reasons.

- A. `printMyData( "Kusal" , "Nimal"`)
- B. `printMyData("Natasha" , 50.0)`
- C. `printMyData( 25 , 15.0)`

[15 marks]

c. Consider the below given *Fish*, *GoldFish* and the *Demo* classes.

<pre>public class Fish {     public static int fins = 5;     public String color = "White";      public String swim(){         return ("Fish is Swimming");     } }</pre>	<pre>public class GoldFish extends Fish {     public int pairedFins = 2;     public String color = "Gold";      public String swim(){         return ("Gold Fish is Swimming");     } }</pre>
<pre>public class Demo{     public static void main(String[] args){         GoldFish g = new GoldFish();         Fish f = new GoldFish();         //Code Line 05     } }</pre>	

Write down the output when compile and run Demo class by replacing Code Line 05 with following Java statements. Give your reasons for each scenario.

- A. System.out.println( *Fish.fins* );
- B. System.out.println( *g.fins* );
- C. System.out.println( *g.swim()* );
- D. System.out.println( *f.swim()* );
- E. System.out.println( *g.color* );
- F. System.out.println( *f.color* );
- G. System.out.println( *f.pairedFins* );
- H. System.out.println( *g.pairedFins* );

[40 marks]

3.

a. *Encapsulation is a mechanism of wrapping the data and code acting on the data (methods) together as a single unit.*

i. Write down *one (01)* advantage and *one (01)* disadvantage of Java Encapsulation.

[10 marks]

ii. Encapsulate the given *Student* class below by making necessary changes to access and modify the *age* field only through *accessors* and *mutators*.

```
public class Student {
    protected int age;
}
```

[10 marks]

b. Consider the below given *MyRunnable* interface.

```
public interface MyRunnable {
    void print(int x){
        System.out.println(x);
    }
}
```

i. What is the problem associated with the *MyRunnable* interface.

[05 marks]

ii. Re write the *MyRunnable* interface in two ways by solving the problem you have identified above. (Without changing *MyRunnable* interface as a class or an abstract class.)

[10 marks]

iii. Write down the purpose of using an interface like *Tager* shown below in Java.

```
public interface Tager {
}
```

[05 marks]

c. *Abstract class is a restricted class that cannot be used to create objects.*

i. Can an abstract class have a constructor? If yes write down *one (01)* usage of it.

[10 marks]

ii. Write down *two (02)* differences between *Abstract Classes* and *Concrete Classes*.

[10 marks]

d. *Exception is an event that disrupts the normal flow of the program.*

i. By giving an example, briefly describe the concept of *Unchecked Exceptions*.

[10 marks]

ii. Use the following *ExceptionDemoto* Java classes to answer the questions given below.

```
public class ExceptionDemo{
    public static void main(String[] args){
        ExceptionDemo ed = new ExceptionDemo();
        ed.divTwoNums(10, 0);
    }

    public void divTwoNums(int x, int y){
        System.out.println ("Begin" );

        int z = x / y ; //    -> div statement

        try{
            System.out.print ("1" );
            // A
        }catch(Exception e){
            System.out.print ("2" );
        }
        System.out.println ("End" );
    }
}
```

- A. What will be the output when you compile and run the above *ExceptionDemo* Java class. Justify your answer.
- B. What will be the output when you move *div statement* to *//A* and compile and run the above *ExceptionDemo* Java class. Justify your answer.
- C. What will be the output when you move *div statement* to *//A* and replace the catch block with following try block and compile and run the above *ExceptionDemo* Java class. Justify your answer.

```
finally{
    System.out.print ("3");
}
```

[30 marks]

4.

a. *JDBC Driver is a software component that enables java application to interact with the database.*

- i. Write down *two (02)* types of Java Database Connectivity (JDBC) drivers.

[10 marks]

- ii. Complete the following statement to create a Connection object by using the given connection details below.

*Host: 127.0.01*

*Port No: 3306*

*DBMS: MYSQL*

*Database Name: ictec*

*User Name: root*

*Password: 1234*

Statement:

*Connection con = DriverManager. //Youranswer*

[10 marks]

b. *All Java programs have at least one thread.*

- i. Write down the *five (05)* major stages of the *life cycle of a Thread*.

[10 marks]

- ii. Consider the *MyThread* and *Demo* Java classes given below.

<pre>public class MyThread implements Runnable {     public void run()     {         String name = Thread.currentThread().getName();         System.out.println("Hello by "+ name );     } }</pre>	<pre>public class Demo {     public static void main(String[] args)     {         MyThread mt = new MyThread();         mt.setName("Perera");         mt.start();     } }</pre>
--	---

- A. Write down the output when you compile and run the above java program. Justify your answer.

[15 marks]

- B. What are the code changes to be done to get *Hello by Kamal* as the output.

[15 marks]

c. Complete the *MyFileRW* class by full filling the given requirements.

Requirements:

- Purpose is to Read and Write user entered statement to a text(.txt) file
- File name: "UserData.txt"
- File location: "User" folder in "D" drive
- User entered data must append to the end of the existing data in the file
- To write: The class uses "DataOutputStream"
- To Read: The class uses "DataInputStream"
- *MyFileRW* class uses "Buffers" to enhance the quality of the writing and reading operations

Note:

- o Assume that user entered data is already stored in a String variable named *uData*
- o Assume that you are using same *UserData* file for both Write and Read operations
- o Assume that you have already imported all the required libraries.

```
public class MyFileRW {  
    public static void main(String[] args) {  
        // code part handling to get the user input  
        // assume this part is already completed and data is stored in uData  
        try {  
            // Your code to create the file  
  
            // Your code to complete the writing part  
  
            // Your code to complete the reading part  
  
        } catch (FileNotFoundException ex) {  
            System.out.println("File not found");  
        } catch (IOException ex) {  
            System.out.println("IOException occurred");  
        }  
    }  
}
```

[40 marks]

//////////////////////////////////////\*\*\*End of Paper\*\*\*//////////////////////////////////////