University of Ruhuna - Faculty of Technology

Bachelor of Information & Communication Technology Honours Degree
Level 2 (Semester I) Semester Examination, June/July 2025
Academic year 2023/2024

Course Unit: ICT 2122 - Object Oriented Programming (Written) **Duration: 02hrs**

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. *A Java program is a set of classes that define objects which interact with each other by calling methods to perform tasks and solve problems.*

i. Briefly describe the following concepts associated with Java Objects.

- ***State***
- ***Behavior***
- ***Identity***
- ***Type***

ii. Consider the method declaration given below. Fill out the following table by considering the parts of the method declaration.

```
public final int getDivision ( int a, int b) throws ArithmeticException {
    return a / b;
}
```

Element	Value in *getDivision* method	Required Always (Yes/No)
Access modifier		
Optional specifier		
Return type		
Method name		
Parameter list		
Exception list		
Method body		

[26 marks]

b. Consider the given ***SwitchTest*** class below.

```
public class SwitchTest {
   public static void main(String[] args) {
      System.out.println("Result: " + test());
   }

   public static String test() {
      for (int i = 1; i <= 3; i++) {
         switch ( i ) {
            case 1: System.out.println("One");
                    break;
            case 2: System.out.println("Two");
                    continue;
            case 3: System.out.println("Three");
                    return "Stopped at Three";
         }
         System.out.println("End " + i);
      }
      return "Completed";
   }
}
```

i. What does the ***break*** statement in **case 1** do in this code?

ii. What happens when the ***continue*** statement is executed in ***case 2*** ?

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

[24 marks]

c. i. List down ***two (02)*** usages of ***parameterized constructors*** used in Java classes.

ii. Consider the given ***Car, Vezel*** and ***Test*** classes below.

```
public class Car {
   public Car() {
      System.out.println("A");
   }
}
```

```
public class Vezel extends Car {
   public Vezel() {
      this(5);
      System.out.println("B");
   }

   public Vezel(int x) {
      super();
      System.out.println("B para : " + x);
   }
}
```

```
public class Test {
   public static void main(String[] args) {
      new Vezel();
   }
}
```

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

[25 marks]

d. Consider the given ***MyInitilizerTest*** classe below.

```
public class MyInitilizerTest {
   static int staticVar = print("x");
   int instanceVar = print("a");

   static {
      print("y");
   }

   {
      print("b");
   }

   MyInitilizerTest() {
      print("c");
   }

   static int print(String msg) {
      System.out.println(msg);
      return 0;
   }

   public static void main(String[] args) {
      System.out.println("z");
      new MyInitilizerTest();
      new MyInitilizerTest();
   }
}
```

i. List down ***two (02)*** usages of ***static initializers*** used in Java classes.

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

[25 marks]

2. a. Consider the ***MyDemo*** class given below

```
public class MyDemo {
    public static void main(String[] args) {
        int a = 100, b = 200;

        Integer p, q, x, y;

            p = a;
            q = a;          Section A

            x = b;
            y = b;

        System.out.println(p == q);
        System.out.println(x == y);
        System.out.println(x.equals(y));
    }
}
```

i. What is the Java concept used in ***section A*** of the above ***MyDemo*** class?

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

[20 marks]

b. Consider the ***Animal, Dog*** and ***Demo*** classes given below.

```
public class Animal {

    public Animal getAnimal() {
        System.out.println("Animal ");
        return new Animal();
    }
}
```

```
public class Dog extends Animal {

    public Dog getAnimal() {
        System.out.println("Dog ");
        return new Dog();
    }
}
```

```
public class Demo {
    public static void main(String[] args) {
        Animal a = new Animal();
        Animal d = new Dog();

        a.getAnimal();
        d.getAnimal();
    }
}
```

i. What are the OOP concepts used in the above java classes?

ii. What will be the ***output*** when you compile and run the above ***Demo*** class? Give your reasons for each output value.

[20 marks]

c. State whether you agree with the following statements. Give your reasons for each.

i. *final methods execute more efficiently than nonfinal methods in Java*

ii. *private methods are automatically considered to be final in Java*

[20 marks]

d. Consider the ***vehicle, Truck*** and ***Demo*** classes given below.

```
public abstract class Vehicle {
   public static int wheels = 4;
   public String color = "White";

   public static String turn(){
      return ("Vehicle is Turning");
   }

   public String drive(){
      return ("Vehicle is Moving");
   }
}
```

```
public class Truck extends Vehicle {
   public static int wheels = 6;
   public String color = "Yellow";

   public static String turn (){
      return ("Truck is Turning ");
   }

   public String drive(){
      return ("Truck is Moving ");
   }
}
```

```
public class Demo{
   public static void main(String[] args){
      Truck t = new Truck();
      Vehicle v = new Truck();
      //Code Line 05
   }
}
```

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

i. System.out.println(***t.wheels***)

ii. System.out.println(***v.wheels***);

iii. System.out.println(***t.color)***);

iv. System.out.println(***v.color***);

v. System.out.println(***t.turn()***);

vi. System.out.println(***v.turn()***);

vii. System.out.println(***t.drive()***);

viii. System.out.println(***v.drive()***)

[40 marks]

3. a. Consider the ***Student*** class given below.

```
public class Student {
    private String name;
    private int age;

    public void setDetails(String n, int a) {
        name = n;
        age = a;
    }

    public void showDetails() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}
```

i. What ***key OOP concept*** is demonstrated in the above Java code?

ii. Briefly explain how the concept is implemented in the above given ***Student*** class.

iii. Write down ***one (01) advantage*** and ***one (01) disadvantage*** of using that OOP concept in Java programming.

[30 marks]

b. Consider the two interfaces ***IntA, IntB*** and ***MyClass*** class given below.

```
public interface IntA {
    int MYVAL = 100;

    default void show() {
        System.out.println("Default method...");
    }
}
```

```
public interface IntB {
    int MYVAL = 200;

    static void display() {
        System.out.println("Static method...");
    }
}
```

```
public class MyClass implements IntA, IntB {

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

}
```

i. Write down the Java codes you need to achieve following requirements in ***//Your Code area***.

- To create an *object* of ***MyClass***.
- To call the ***show()*** method from IntA.
- To call the ***static method display()*** from IntB.
- To display ***MYVAL*** from both interfaces.

ii. Do you agree with the following statement? Give your reasons.

In Java, it is allowed to implement a Java interface from another interface

[30 marks]

c. Consider the ***InvalidNameException*** and ***ExceptionDemo*** classes given below.

```
public class InvalidNameException extends Exception {
    public InvalidNameException(String message) {
        super(message);
    }
}
```

```
public class ExceptionDemo {
    public static void main(String[] args) {
        try {
            readFile("data.txt");
            divide(10, 0);
            validateName("");
        } catch (Exception e) {
            System.out.println("Exception Occured: " + e.getMessage());
        }
    }

    public static void readFile(String filename) throws IOException {
        FileReader fr = new FileReader(filename);
        fr.close();
    }

    public static void divide(int a, int b) {
        System.out.println(a / b);
    }

    public static void validateName(String name) throws InvalidNameException {
        if (name.isEmpty()) {
            throw new InvalidNameException("Name cannot be empty.");
        }
    }
}
```

i. Identify the ***checked exception*** used in the above code.
Briefly explain why it is classified as a checked exception.

ii. Write down the possible java statement in the code which might throw an ***unchecked exception***?
What is the type of unchecked exception it might throw?

iii. What is the purpose of the ***InvalidNameException*** class in the above given code?

iv. Is it necessary to declare ***throws IOException*** in the ***readFile*** method?
Give your reasons.

v. What will be the output when you compile and run ***ExceptionDemo*** class after ***removing the try-catch block*** in the main method?
Give your reasons.

[40 marks]

4. a. *Generics and collections in Java allow you to store and manage groups of objects in a type-safe and reusable way.*

i. Create a generic class called ***ItemStore<T>*** that can store any type of item.
For that your class must
- Have a private variable of type T named **item**.
- Include a method **set(T item)** to ***store a value***.

- Include a method get() to ***retrieve the stored value.***

*Then, in the main method in the same **ItemStore** class write statements to:*

- Create an itemStore to ***store an integer*** value.
- Create an ItemStore to ***store a String*** value.
- Print both stored values to the ***console.***

[20 marks]

ii. Write a Java program that can ***store any number of names*** (in Strings) using your knowledge in Java Collections.

- Add the names "*Nimal*", "*Kamala*", and "*Santush*" to your collection.
- Print each name on a new line in the console.

[20 marks]

b. Consider the ***MyThread*** and ***Demo*** Java classes given below.

```
public class MyThread extends Thread {
    public void run()
    {
        String name = Thread.currentThread().getName();
        System.out.println("Hello from "+ name );
    }
}
```

```
public class Demo {
    public static void main(String[] args)
    {
        MyThread mt = new MyThread();
        Thread rt = new Thread(mt);
        rt.setName("Anura");
        rt.start();
    }
}
```

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

ii. What are the code changes to be done to get ***Hello from Sajith*** as the output using the same *mt* thread above.

[20 marks]

c. Consider the ***DbConnect*** Java class that connects to a MySQL database.

```
public class DbConnect {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/fot";
        String user = "root ";
        String password = "1234";

        try {
            Connection conn = DriverManager.getConnection(url, user, password);
        } catch (SQLException e) {
            System.out.println("Connection failed!");
        }
    }
}
```

Refactor the above ***DbConnect*** class to use the Singleton Design Pattern by ensuring that:

- The singleton instance is created at class loading time.
- The class maintains *only one JDBC connection*
- The class provides a public method to access the Connection object.

[40 marks]