

**Course Title:** Object Oriented Programming – Java I.

**Content:** Examination Marking Guide

### Instructions

1. Answer question **ONE (Compulsory)** and any other **TWO** questions.
2. Write all your answers in the answer booklet provided.
3. Time allowed: **Two hours**.

### SECTION ONE

#### QUESTION ONE [30 Marks]

a) State and explain any three features that characterize a programming language as Object Oriented. [6 Marks]

The use of classes and objects, Inheritance, Encapsulation, Polymorphism etc

b) Using an appropriate example, explain how encapsulation is achieved in Java. [5 Marks]

- Data hiding using encapsulation
  - Data fields are usually private
  - The client application accesses them only through public interfaces
- set method
  - Controls the data values used to set a variable
- get method
  - Controls how a value is retrieved

c) State and explain the roles of two instance method types that can be defined in Java classes [6 Marks]

The student should differentiate between accessor and mutator methods

d) Using Java, demonstrate how to create an array containing five Kenyan universities: Maseno, Kenyatta, Nairobi, Moi and Egerton. Then further explain how you could use a while loop to display all university names in the array the array. [6 Marks]

Creation of an array

```
String[] University = {"ANU", "MMU", "TUK", "MUA", "SPU"};
```

Accessing array elements:

```
int i=0;
while(i<4){
    System.out.println(University[i]);
```

```
    i++;  
}
```

**e) Using an appropriate example, explain polymorphism as used in Object Oriented**

**Programming.**

**[7 Marks]**

Using the same method name to indicate different implementations is called polymorphism, a term that means “many forms”—many different forms of action take place, even though you use the same word to describe the action. In other words, many forms of the same word exist, depending on the object associated with the word.

## **SECTION TWO**

### **QUESTION TWO [20 Marks]**

**a) State and explain any four differences between Object Oriented Programming and**

**Procedural/Imperative Programming.**

**[8 Marks]**

\*the students should bring out aspects like the use of: classes, objects, inheritance, encapsulation, polymorphism, abstraction among others.

**b) Consider exceptions in Java:**

**i) What four types of exceptions can occur in a Java program?**

**[4 Marks]**

The programs you write can generate many types of potential exceptions, such as when you do the following:

- You issue a command to read a file from a disk, but the file does not exist there.
- You attempt to write data to a disk, but the disk is full or unformatted.
- Your program asks for user input, but the user enters invalid data.
- The program attempts to divide a value by 0, access an array with a subscript that is too large, or calculate a value that is too large for the answer’s variable type.

**ii) Using an appropriate example, discuss the mechanism used by Java to handle any one of the above mentioned exceptions**

**[8 Marks]**

\*the student should demonstrate how the try ... catch ... finally exception handling block works. The example should capture the correct exception type.

### **QUESTION THREE [20 Marks]**

**a) Write a program that prompts for name and year of birth from the console then computes the age. Let the program display the name and age on the console as it stores the same in a text file called “ages”.**

**[12 Marks]**

\*the students should bring out proper usage of console prompts and writing to text files.

**b) Consider a program that prompts for two integer values from the console then stores them in two variables x and y. It should then divide x by y as it stores the results in variable z.**

i) **What to potential reasons could make such a program to crush at runtime? [2**

**Marks]**

- Division by zero
- Input type mismatch

ii) **Write the program with proper mechanisms that take care of the two crashes. [6**

**Marks]**

```
import java.util.InputMismatchException;
import java.util.Scanner;
```

```
public class FirstProject {
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
        int numerator, denominator, result;
        try{
            System.out.print("Enter numerator >> ");
            numerator = input.nextInt();
            System.out.print("Enter denominator >> ");
            denominator = input.nextInt();
            result = numerator / denominator;
            System.out.println(numerator + " / " + denominator + " = " + result);
        }
        catch(ArithmeticException mistake){
            System.out.println(mistake.getMessage());
        }
        catch(InputMismatchException mistake){
            System.out.println("Wrong data type");
        }
    }
}
```

#### QUESTION FOUR [20 Marks]

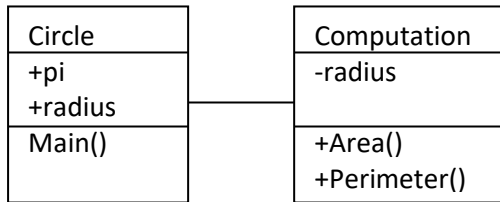
a) Consider a Java program with two value-returning methods `area()` and `perimeter()`. Let the program prompt the user for the length and width of a rectangle through `main()`. It should then compute the area and perimeter from respective methods. The program should then display the values returned to `main()`. Let the output be as follows.

```
Enter Length:
Enter Width:
The area is ___ cm2
The perimeter is ___ cm
```

**[10 Marks]**

\*proper creation of methods, passing of parameters and display of output will be considered.

b) Consider the following UML diagram



Write a java program that implements the model

[10 Marks]

```
public class Circle {
    public static void main(String[] args) {

        final double PI=3.142;
        int radius;
        radius=3;

        Circle circleobj=new Circle();
        circleobj.computearea(PI,radius);
        circleobj.computeperimeter(PI,radius);

        System.out.println("The Area is " + circleobj.printarea() + "
        The Perimeter is " + circleobj.printperimeter());
    }
}

public class computation {

    private double area, perimeter;

    public double printarea(){
        return area;
    }

    public double printperimeter(){
        return perimeter;
    }

    public void computearea(double pi, int rad){
        area=pi*rad*rad;
    }

    public void computeperimeter(double pi, int rad){
        perimeter=pi*(rad*2);
    }
}
```

#### QUESTION FIVE [20 Marks]

a) Three employees in a company are up for a special pay increase. You are given a file, say EmployeeData.txt with the following data:

Miller Andrew 10000 5.0  
 Green Sheila 12000 6.0  
 Omollo John 35000 1.0

Each input line consists of employee's last name, first name and current salary and percentage increase. For example, in the first input line, the last name of the employee is Miller, first name is Andrew, the current salary is 10000 and the pay increase is 5.0%. Write a program that reads data from the specified file and stores the output in another file EmployeeOutput.txt. For each employee, the data must be output in the following format: **firstname lastname updatedsalary.** [10 Marks]

```
package today;

import java.io.*;
import java.util.*;

public class Today {

    public static void main(String[] args) throws FileNotFoundException
    {
        String firstname;
        String lastname;

        int salary;
        double rate;

        double updatedsalary;

        PrintWriter writeinfile = new PrintWriter("D:\\input.txt");

        writeinfile.println("Miller Andrew 10000 5.0");
        writeinfile.println("Green Sheila 12000 6.0");
        writeinfile.println("Omollo John 35000 1.0");

        writeinfile.close();

        Scanner readinfile = new Scanner(new FileReader("D:\\input.txt"));
        lastname=readinfile.next();
        firstname=readinfile.next();
        salary=readinfile.nextInt();
        rate=readinfile.nextDouble();

        updatedsalary=salary+(salary*rate/100);

        readinfile.close();

        PrintWriter writeoutfile = new PrintWriter("D:\\output.txt");
        writeoutfile.println(firstname + " " + lastname + " " + updatedsalary);

        writeoutfile.close();
    }
}
```

- b) Create a class named Employee with a name and a salary. Make a class named Manager that inherits from Employee with an instance field named department. Supply a toString() method that prints the manager's name, department, and salary. Make another class named Director that inherits from Manager with an instance field named stipendAmount. Supply the toString() method for Director that prints all of its instance variables. Also, write a program named myOutput that instantiates an object of each of the classes and invokes the toString() method of each of the objects. [10 Marks]

```
class Employee
{
    private String name = "John Doe";
    private int Salary = 2000;
```

```
    public String toString()
    {
        return "Name: " + name + "\nSalary: " + salary;
    }
}

class Manager extends Employee
{
    private String departmentName = "College of Arts and Science";

    public String toString()
    {
        return super.toString() + "\nDepartment Name: " +
departmentName;
    }
}

class Director extends Manager
{
    private int stipendAmount = 800;

    public String toString()
    {
        return "Stipend amount is: " + stipendAmount;
    }
}

public class myOutput
{
    public static void main(String args[])
    {
        Manager managerOut = new Manager();
        System.out.println(managerOut.toString());
        Director directorOut = new Director();
        System.out.println(directorOut.toString());
    }
}
```