CHUKA  **UNIVERSITY**

UNIVERSITY EXAMINATIONS

RESIT/SPECIAL EXAMINATION

SECOND YEAR SEMESTER TWO EXAMINATION FOR BACHELOR OF SCIENCE IN COMPUTER SCIENCE

RESIT EXAM

COSC 223: OBJECT ORIENTED PROGRAMMING

STREAMS (BSC. COMP SCIENCE) **TIME: 2 HOURS**

DAY/DATE: MONDAY 16/11/2020 **8.30 A.M. – 10.30 A.M.**

INSTRUCTIONS: Answer Question One and any other two questions

QUESTION ONE (30 MARKS)

(a) Explain why it is considered good practice to limit the scope of fields and methods in object oriented programming. (3 marks)

(b) Identify three differences between a constructor and a method. (3 marks)

(c) Explain what encapsulation is and show how it is achieved in Java. (4 marks)

(d) Show the output when the following code executes. Explain. (4 marks)

```
int y = 53, count = 7;
while (true){
     System.out.println("Y value: " + y);
     count = y % count;
     y += count;
     if (count > 100 ){
         break;
     }
}
```

(e) Show the expected print out when the program below executes. (4 marks)

```
class A
{
    public A()
    {
        super();
        System.out.println("Class A");
    }
   void print(){
        System.out.println("Hello in A");
    }

}

class B extends A
{
    public B()
    {
        super();
        System.out.println("Class B");
    }
   void print(){
        System.out.println("Hello in B");
    }

}

class C extends B
{
    public C()
    {
        super();
        System.out.println("Class C");
    }
   void print(){
         System.out.println("Hello in C");
         super.print();
    }

}



public class MainClass
{
```

```
    public static void main(String[] args)
    {
        C c = new C();
        c.print();
    }
}
```

(f) Declare and initialize an array of six string values to capture the following first names: `Joy`, `Jolly`, `Fred`, `John`, `James` and `Carlos`. (4 marks)

(g) Show the output expected when the following piece of code executes. (4 marks)

```
public class RTException {
    public int add (int a, int b){
        if(b == 0){
             throw new ArithmeticException("Illegal attempt to divide by zero.");
        }
       return a+b;
    }
    public static void main(String [] args)
    {
        RTException rtObj = new RTException();
        try {
            System.out.print("Hello inputs! \n");
            int x =rtObj.add(40,0);
            System.out.println("Value of x: " +x);
            System.out.print("Hello again! \n");
        }
        catch (ArrayIndexOutOfBoundsException re ) {
            System.out.println("Error: " + " problem caught \n");
        }
        catch (ArithmeticException re ){
            System.out.println("Error: " + re.getMessage());
        }
        finally {
            System.out.println("Check results of exception handling. ");
        }
        System.out.println("Hello again! ");
    }
}
```

(h) Consider the Java code segment shown below. Identify local variables, instance variables and class variables and instance methods used in the code. (4 marks)

```
public class Student{
  private string firstName;
  private string lastName;
  private date dateOfBirth;
  public static string programme;

  public Student (){
```

```
        this.firstname = null;
        this.lastName = null;
        this.dateOfBirth = null;
        }

    Public updateDetails(string fname, string lname, date dob){
        this.firstname = fname;
        this.lastName = lname;
        this.dateOfBirth = dob;
        }

    Public void getAddress(/* address details */){
        //get address details
    }
    Public static void main(){
        Student.programme = "Bsc Computer Science";
        Student paul = new Student();
        paul.updateDetails("Paul", "McBrian", 1996);

    }
```

QUESTION TWO (20 MARKS)

(a) Explain the difference between method overloading and method overriding. (4 marks)

(b) Consider the following code in Object Oriented Programming. it defines the start of a class to represent bank accounts:

```
public class BankAccount{
  private int interest_rate = 0.3;
  private double balance;
  private String name;
  public BankAccount(String name, int number, double balance){
        this.name = name;
        this.number = number;
        this.balance = balance;
  }
```

(i) Add instance methods `deposit()` and `withdraw()` which increase and decrease the balance of the account. Make sure the `withdraw()` method doesn't allow the account to go into overdraft. Add a third method called `addInterest()` which adds interest to the balance (the interest should be the interest rate multiplied by the current balance). (8 marks)

(ii) Create a subclass of `BankAccount` called `StudentAccount`. Every `StudentAccount` should have an overdraft limit of Kshs 10,000. Write a constructor for the new class. Override the `withdraw()` method to make sure that students can withdraw money up to their overdraft limits. (8 marks)

QUESTION THREE (20 MARKS)

(a) Consider the Java program shown below:

```
public class A
{
     private int a;
     protected int b;
     public int c;
     public A();
     public void seta(int new_a);
     public void setb(int new_b);
     public void setc(int new_c);
     public int geta();
     public int getb();
     public int getc();
}
public class B extends A
{
     private int d;
     public B();
     public void setd(int new_d);
     public int getd();
}
```

(i) What is the relationship between the two classes? Explain. (4 marks)

(ii) Identify the members of B. Explain your reasoning. (6 marks)

(b) Write a Java method for finding the average of an array of scores passed to the method. The method returns the average. If the array has no scores or the access to the scores goes beyond the array size, throw a new exception in the method. (10 marks)

QUESTION FOUR (20 MARKS)

Study the calculator program shown.

(a) Identify the errors in the program shown below. Rewrite the program correcting all the errors. Attach single line comment to show the error that was made. (10 marks)

(b) Identify exception propagators and catchers in the corrected code. (4 marks)

(c) What is the expected use of the keyword "this" as used in method getX and getY. (3 marks)

(d) Explain the role of the methods: getX, getY and main. (3 marks)

```
import java.io.*;
public class Cal culator1 {
    private int x;
    private int y;
    public int getX(String x){
        return this.x = Integer.parseInt(x);
    }
    public int getY(String y){
        return this.y = Integer.parseInt(y);
    }
    public add(){
        return x+y;
    }
    public subtract (){
        return x-y;
    }
    public int multiply(){
        return x*y;
    }
    public int remainder throws Exception
        if (y == 0){
            throw Exception("Divide by zero");
        }
        return x%y;
    }
    public showResult()throws Exception{
        System.out.println("X = " + x);
        System.out.println("Y = " + y);
        System.out.println("X + Y: " + add());
        System.out.println("X - Y: " + subtract());
        System.out.println("X * Y: " + multiply());
        System.out.println("X / Y: " + remainder());
    }

    public static void main(String[] args) throws Exception{
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
```

```
        Calculator1 calc=new Calculator1();

            System.out.print("Enter value of x: ");
            calc.getX(in
            System.out.print("Enter value of y: ");
            calc.getY(in.readLine());
            calc.showResult();
        }
        catch (Exception e){
         System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Question Five (20 marks)

(a) Explain what an Interface is. Use Java code sample to show the syntax of an interface in Java and show how an interface is used in Java. (4 marks)

(b) Consider a GUI interface given below.

(i) Discuss the possible Swing components used in the GUI and the order of their insertion to get a similar look. (12 marks)

(ii) Identify the necessary interface that ought to be implemented and which method ought to be overridden. Give an illustration using Java code sample. (4 marks)