04/06/2021

COM 2044 Final Exam

OBJECT ORIENTED PROGRAMMING

Name:

Number:

Duration: 80 Minutes

Q-1: (20p) Write the screen output that results from the run of the codes given below. (Note: Write the output completely.)

```
package final_exam;
public abstract class abstractClass {
    private int num1;
    private int num2;

    public abstractClass(int num1, int num2)
    {
        this.num1=num1;
        this.num2=num2;
    }

    public int getNum1() {
        return num1;
    }

    public void setNum1(int num1) {
        this.num1 = num1;
    }

    public int getNum2() {
        return num2;
    }

    public void setNum2(int num2) {
        this.num2 = num2;
    }

    public abstract void calculate(int num1, int num2);
}
```

```java
package final_exam;
public class class_1 extends abstractClass{
    public class_1(int num1, int num2) {
        super(num1, num2);
    }

    @Override
    public void calculate(int num1, int num2) {
      System.out.print("I am class_1. My result is ");
      System.out.println((2*((num1+this.getNum1())+(num2+this.getNum2()))))+".");
    }
}
```

```java
package final_exam;
public class class_2 extends abstractClass{
    public class_2(int num1, int num2) {
        super(num1, num2);
    }

    @Override
    public void calculate(int num1, int num2) {
      System.out.print("I am class_2. My result is ");
      System.out.println((2+(this.getNum1()*this.getNum2()))+".");
    }
}
```

```java
package final_exam;
public class Final_exam {
    public static void main(String[] args) {
        class_1 c1=new class_1(4,4);
        class_2 c2=new class_2(3,5);
        c1.setNum1(3);
        c2.setNum2(4);
        c1.calculate(1, 2);
        c2.calculate(1, 2);
    }
}
```

Q-2: (30p) It is expected you to define a simple Java application code that displays two buttons (Figure 1), and responds to the user click by displaying an appropriate dialog box (Figure 2). However, this example can be extended to much more complicated user interfaces. It is expected you to implement this class in two different ways - the single listener approach, and the inner class approach.

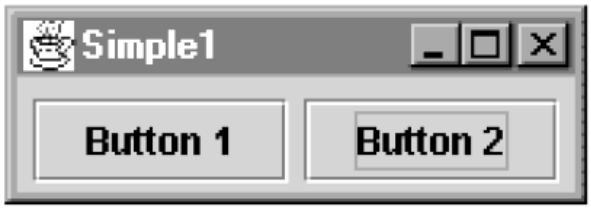

Figure 1: Simple Swing Application

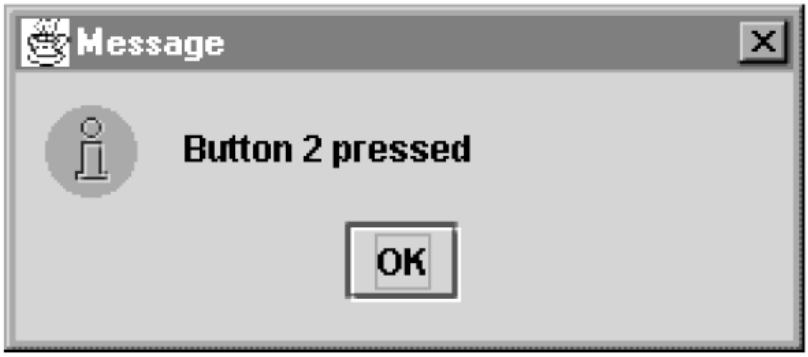

Figure 2: Dialog Window - response to event

Q-3: (20p) Draw a Use Case Diagram for a voting machine, on which voters can see a list of candidates and select one to vote for. The machine should check that each voter is eligible to vote. The electoral registrar will also want to print a summary of the total votes for each candidate, and (separately) a list of the voters who have voted, and a list of those who haven't. In case of a dispute, the machine should also list a complete record of who voted for whom, but only a judge can use this function.

Q-4: (30p) A factory has different kind of robots for constructing cars. A paint robot paints the car, a configuration robot configures the car from given parts, and a testing robot tests the car on possible errors. Each robot can be put on and off, and can move to a given coordinate.

a-) (15p) Give a class diagram for this application:
b-) (15p) Convert this class diagram into java code: