CS390 – FPP EXAM-REVIEW

Midterm - Saturday – 9.45 - 12 noon (3/5/2022)

Lessons for Examination

Lesson – 3 - Objects and Classes

Lesson – 4 - Recursion

Lesson – 5 – Inheritance, Interface, and Polymorphism

Lesson – 6 & 7 – Sorting or Listener Implementation for the Button click Events using Inner class implementation.

Course Resources : www.online.cs.miu.edu

Reading Resources : Demo Code and Homework Solutions

Important points for the FPP Exam

1. The midterm will be held on 3/5/2022 – Saturday Morning a V32 in-person and paper exam.
2. The examination will be timed. It will begin at 9.45 am and will end at 12:00 noon.
3. Examination should be closed book.
4. Mobile should be in Silent or Swich off mode. You are not allowed to keep the mobile and smart things(like smart watch). So bring backpack to keep your belongings. Keep the backpack infront of the dias.
5. The midterm contains four programms from Lesson 3 – Lesosn 7.
6. Kindly follow the academic honesty.
7. Try to avoid rest room break.

   Program – 1- Lesson-3

   a. Class, Objects & Constructors, Object methods toString() and equals().
      a. Needs to write a Test/Main class for the given class and perform some computation proceessing the collection of array objects.
      b. Able to know how to create instance and static methods.
      c. Able to know data types, decision making, looping, switch and Enum.
      d. Able to know how to override toString(), equals()

Program – 2- Lesson-4

a. Need to write a recursive method for the given problem using base case and recursive case. (Refer String examples, searching algorithm and mathematical examples discussed in that lesson and your homework problems)

Program – 3 - Lesson-5 – Partial Coding. Able to know how to work with the below concepts (a) to (d).

a. Inheritance

b. Abstract class & Polymorphism –Partial coding

c. Interface & Polymorphism –Partial coding

d. Convert Non-OO code to OO-Code

Program – 4 – Lesson 6 & 7 – Partial Coding

a. Able to know how to perform Button click Listener Implementation using the Inner class concepts. Choice is up to you. You can implement using Member Inner class / Local Inner class, Anonymous Inner class / Lambdas. No need to worry about designing UI part. You are going to implement only the listner implementation part.

(or)

b. Able to perform sorting for the given array collections using Comparator implement with the Inner class concepts.

Sample Problems:

Lesson – 3

In the test class able to know how to decalre an array of objects, how loop through the array and perform some computation. Able to overide object methods.

```
public class TrailorRental {
        private double mRent;
        private String tNo;
        TrailorRental (String tNo, double mRent)
        {
                this.tNo= tNo;
                this.mRent = mRent;
        }
        public double getmRent() {
                return mRent;
        }
        public void setmRent(double mRent) {
                this.mRent = mRent;
```

```
    }
    public String toString() {
        return "[" + this.tNo + "," + this.mRent + "]";
    }
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        TrailorRental other = (TrailorRental) obj;
        if(this.mRent==other.mRent && this.tNo==other.tNo)
        return true;
        else
            return false;
    }

}
public class TestTrailerRental {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        TrailorRental[] obj = { new TrailorRental("11A",650),
                                new TrailorRental("10J",550),
                                new TrailorRental("16D",750)};
        System.out.println("Original Rental for the Trailors : " );
        for(TrailorRental t: obj)
            System.out.println(t);
        changeRental(obj);
        System.out.println("\nModified Rental for the Trailors : " );
        for(TrailorRental t: obj)
            System.out.println(t);
        // Usage of equals() and hashCode()
        TrailorRental t1 = new TrailorRental("11A",650);
        TrailorRental t2 = new TrailorRental("11A",650);
        TrailorRental t3 = new TrailorRental("12D",750);
        System.out.println(t1.equals(t2));

    }
    public static void changeRental(TrailorRental[] objects)
    {
        double temp =0.0;
        if (objects!= null || objects.length >0)
        {
            for(TrailorRental ind : objects)
            {
                temp = ind.getmRent() + ind.getmRent() * 0.10;
                ind.setmRent(temp);
            }

        }
    }
}
```

Lesson – 4 – Recursion

a. The two criteria for correct recursion. (Base case, and progress to the base case(Recursive Case))

Using Recursion to count the number of vowels in a string.

```
public class CountVowel {

    public static void main(String[] args)
    {
        String string;
        Scanner in=new Scanner(System.in);
        System.out.print("Enter any Word : ");
        string=in.next();
        System.out.println("Total Vowels in the String " + string + " is:
"+vowelCount(string));
}

    public static int vowelCount(String string){
        char c;
        int count=0;
        if(string.length()==0 || string.equals(“”) || string.equals(null)
            return 0;
        else
        {
            c=string.charAt(0);
            if(c=='a'||c=='e'||c=='i'||c=='o'||c=='u')
                count++;
            return count + vowelCount(string.substring(1));
        }
    }
}
```

Lesson-5

Examples Refer : Homework Problems 1- 5

Demo Code : lesson5.closedcurvebad

lesson5.closedcurcegood

Lesson 6: Refer Sorting Example: Lesson-6 Demo code Packages innersort1, innersort2, … innersort6 and your Homework Solutions

Lesson 7:

Designing code will be given to you. Only concentrate on the following

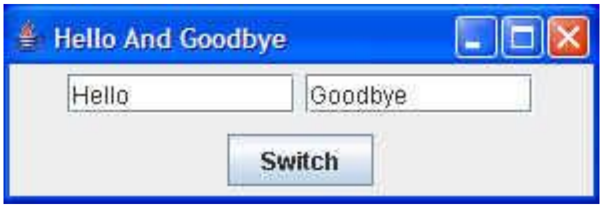
a. Need to practice how to implement ActionListener[Anonymous or Inner or Current class(this) or Lambdas]
b. Need to know how to convert the TextField string input into int, float, double and vice versa.

Click Listener Signature/Syntax will be given to you as below

Syntax of ActionListener

```
public interface ActionListener {
        public void actionPerformed(ActionEvent e);
}
```

Write Swing code which, when run, produces the following screen. Clicking the Switch button causes the values in the two text boxes to be switched (after the click, for example, “Goodbye” appears in the left box and “Hello” in the right box).



<u>Solution</u>

```
public class HelloGoodbye extends JFrame {
        JPanel mainPanel, topPanel, centerPanel;
        JTextField leftText, rightText;

        public HelloGoodbye() {
                setTitle("Hello Goodbye");
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                defineTopPanel();
                defineCenterPanel();
                mainPanel = new JPanel();
                mainPanel.setLayout(new BorderLayout());
                mainPanel.add(topPanel, BorderLayout.NORTH);
                mainPanel.add(centerPanel, BorderLayout.CENTER);
                getContentPane().add(mainPanel);
                pack();
        }

        public void defineTopPanel(){
                topPanel = new JPanel();
                topPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
                leftText = new JTextField(10);
                leftText.setText("Hello");
                rightText = new JTextField(10);
                rightText.setText("Goodbye");
                topPanel.add(leftText);
                topPanel.add(rightText);
        }

// Concentrate on highlighted part

        public void defineCenterPanel() {
                centerPanel = new JPanel();
                centerPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
                JButton b = new JButton("Switch");
                b.addActionListener(new ButtonListener());
                centerPanel.add(b);
        }

        class ButtonListener implements ActionListener {
                public void actionPerformed(ActionEvent evt) {
                    String left = leftText.getText();
                    String right = rightText.getText();
                    leftText.setText(right);
                    rightText.setText(left);
                }
        }

        public static void main(String[] args) {
                            HelloGoodbye hg = new HelloGoodbye();
                    hg.setVisible(true);
        }
}
```