CS390 – FPP EXAM-REVIEW

Midterm - Saturday – 9.45 - 12 noon (6/4/2022)

Lessons for Examination

Lesson – 3 - Objects and Classes

Lesson – 4 - Recursion

Lesson – 5 – Inheritance, Interface, and Polymorphism

Lesson – 6 - Inner Class

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 6/4/2022 – Saturday Morning a V17 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(sum, average, min, max etc.,).
      b. Able to know how to create instance and static methods.
      c. Able to know how to override toString(), equals()
      d. Able to know how to avoid NullPointer Exception by doing proper validation.

Program – 2- Lesson-4

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

b. Your logic does not give StackOverFlowError

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

a. Inheritance – Partial Coding

b. Abstract class & Polymorphism –Partial coding

c. Interface & Polymorphism –Partial coding

d. Convert Non-OO code to OO-Code

Program – 4 – Lesson 6 – Partial Coding

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

b. Practice any one approach for the Comparator Implementation.

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

Daily classnotes: https://sakai.cs.miu.edu/access/content/group/db753b70-c9f9-4de7-9220-297ce59532b3/DailyClassNotes/Lesson-6/

Watch the Recorded Review Video from:

https://mum0.sharepoint.com/sites/CS390-2022-06A-06DRM/Shared%20Documents/VideoSessions/Recordings/Quiz-2%20and%20Midterm%20Review-20220601_134746-Meeting%20Recording.mp4?web=1