# Directions

1. You will complete 6 of the following projects. If you do more than 6, I will grade your best 6.
2. Submit screenshots of working code in this document.
3. Also zip and submit your project folders to Blackboard.
4. Be sure that everything is clearly marked.

# Clarifications

Except for the GUI 1 and GUI 2 the projects can be built using either a console or a JFrame – whichever you prefer

# Choose six projects

Choose six projects that you know that you can complete. Some are easier than others.

The six I chose were the Class,Inherit,String,Array/List,Array of Class, and GUI 2 exercises

## Class

public class Car {  
 private String make;  
 private String model;  
 private String year;  
 private double mileage;  
 private double price;  
  
 public Car(String make, String model, String year, double mileage, double price) {  
 this.make = make;  
 this.model = model;  
 this.year = year;  
 this.mileage = mileage;  
 this.price = price;  
 }  
  
 public Car() {  
 }  
  
 public String getMake() {  
 return make;  
 }  
  
 public void setMake(String make) {  
 this.make = make;  
 }  
  
 public String getModel() {  
 return model;  
 }  
  
 public void setModel(String model) {  
 this.model = model;  
 }  
  
 public String getYear() {  
 return year;  
 }  
  
 public void setYear(String year) {  
 this.year = year;  
 }  
  
 public double getMileage() {  
 return mileage;  
 }  
  
 public void setMileage(double mileage) {  
 this.mileage = mileage;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public void setPrice(double price) {  
 this.price = price;  
 }  
  
 @Override  
 public String toString() {  
 return "Car{" +  
 "make='" + make + '\'' +  
 ", model='" + model + '\'' +  
 ", year='" + year + '\'' +  
 ", mileage=" + mileage +  
 ", price=" + price +  
 '}';  
 }  
  
 public static void main(String[] args) {  
 Car c1 = new Car("Toyota","Corolla","2007",1234.32,4000);  
 Car c2 = new Car("Buick","Lucerne","2007",60000.64,2000);  
  
 System.*out*.println(c1.toString());  
 System.*out*.println(c2.toString());  
 }  
}

A picture containing graphical user interface

Description automatically generated

## Inherit

public class Food extends Product{  
 private String recieved;  
 private String bad;  
  
 public Food(String recieved, String bad,String productName,int quantity,float cost) {  
 super(productName,quantity,cost);  
 this.recieved = recieved;  
 this.bad = bad;  
 }  
  
 public Food() {  
 }  
  
 public String getRecieved() {  
 return recieved;  
 }  
  
 public void setRecieved(String recieved) {  
 this.recieved = recieved;  
 }  
  
 public String getBad() {  
 return bad;  
 }  
  
 public void setBad(String bad) {  
 this.bad = bad;  
 }  
  
 @Override  
 public String toString() {  
 return super.toString() +  
 "recieved='" + recieved + '\'' +  
 ", bad='" + bad + '}';  
 }  
  
 public void displayDayBad()  
 {  
 System.*out*.println("This food goes bad on " + this.bad);  
 }  
  
 public static void main(String[] args) {  
 Food f = new Food("March 1st","March 30","Ketchup",20, 4.99F);  
 f.displayDayBad();  
 System.*out*.println(f.toString());  
 }  
}

Text

Description automatically generated

## String

import java.util.\*;  
public class Strings {  
 static Scanner *sc* = new Scanner(System.*in*);  
  
 public static void main(String[] args) {  
  
 ArrayList<String> wrong = new ArrayList<>();  
 String q1,q2,q3,q4,q5;  
 int correct=0;  
  
 q1 = "What is 1+1?";  
 q2 = "What is 2+2?";  
 q3 = "What is 3+3?";  
 q4 = "What is 4+4?";  
 q5 = "What is 5+5?";  
  
 correct+=*question1*(q1,wrong);  
 correct+=*question2*(q2,wrong);  
 correct+=*question3*(q3,wrong);  
 correct+=*question4*(q4,wrong);  
 correct+=*question5*(q5,wrong);  
  
 System.*out*.println(correct + "/5 correct");  
  
 System.*out*.println(wrong);  
  
 }  
  
 public static int question1(String q1,ArrayList<String> wrong)  
 {  
 System.*out*.println(q1 + "a>1 b>2 c>3 d>4");  
 String ans = *sc*.nextLine();  
  
 if(ans.equalsIgnoreCase("b"))  
 return 1;  
 else  
 wrong.add("Question 1 is wrong");  
 return 0;  
 }  
  
 public static int question2(String q2,ArrayList<String> wrong)  
 {  
 System.*out*.println(q2 + "a>4 b>8 c>12 d>16");  
 String ans = *sc*.nextLine();  
  
 if(ans.equalsIgnoreCase("a"))  
 return 1;  
 else  
 wrong.add("Question 2 is wrong");  
  
 return 0;  
 }  
  
 public static int question3(String q3,ArrayList<String> wrong)  
 {  
 System.*out*.println(q3 + "a>3 b>6 c>9 d>12");  
 String ans = *sc*.nextLine();  
  
 if(ans.equalsIgnoreCase("b"))  
 return 1;  
 else  
 wrong.add("Question 3 is wrong");  
 return 0;  
 }  
  
 public static int question4(String q4,ArrayList<String> wrong)  
 {  
 System.*out*.println(q4 + "a>1 b>2 c>4 d>8");  
 String ans = *sc*.nextLine();  
  
 if(ans.equalsIgnoreCase("d"))  
 return 1;  
 else  
 wrong.add("Question 4 is wrong");  
 return 0;  
 }  
  
 public static int question5(String q5,ArrayList<String> wrong)  
 {  
 System.*out*.println(q5 + "a>5 b>10 c>15 d>20");  
 String ans = *sc*.nextLine();  
  
 if(ans.equalsIgnoreCase("b"))  
 return 1;  
 else  
 wrong.add("Question 5 is wrong");  
 return 0;  
 }  
}

Graphical user interface, text

Description automatically generated

## Array/List

import java.lang.reflect.Array;  
import java.util.ArrayList;  
  
public class NumList {  
 public static void main(String[] args) {  
  
 ArrayList<Integer> nums = new ArrayList<>();  
  
 //Adding nums  
 for(int x=0;x<100;x++)  
 {  
 nums.add((int)(Math.*random*()\*100)+1);  
 }  
  
 System.*out*.println("Average >>> "+ *findAvg*(nums));  
 System.*out*.println("Minimum >>> " + *findMin*(nums));  
 System.*out*.println("Maximum >>> " + *findMax*(nums));  
 System.*out*.println("Standard Deviation >>> " + *findDeviation*(nums));  
 System.*out*.println("Mode >>> " + *findMode*(nums));  
 }  
  
 public static int findAvg(ArrayList<Integer> nums)  
 {  
 int total =0;  
  
 for(int x=0;x<100;x++)  
 {  
 total+=nums.get(x);  
 }  
  
 return total/100;  
 }  
  
 public static int findMin(ArrayList<Integer> nums)  
 {  
 int min = 1000000;  
  
 for(int x=0;x<100;x++)  
 {  
 if(nums.get(x)<min)  
 min = nums.get(x);  
 }  
  
 return min;  
 }  
  
 public static int findMax(ArrayList<Integer> nums)  
 {  
 int max = -1000000;  
  
 for(int x=0;x<100;x++)  
 {  
 if(nums.get(x)>max)  
 max = nums.get(x);  
 }  
  
 return max;  
 }  
  
 public static double findDeviation(ArrayList<Integer> nums)  
 {  
 int avg = *findAvg*(nums);  
  
 double dev = (double)(Math.*sqrt*(avg));  
  
 return dev;  
 }  
  
 public static int findMode(ArrayList<Integer> nums)  
 {  
 int most =0;  
  
  
 for(int x=1;x<=100;x++)  
 {  
 int count =0;  
 for(int y=0;y<100;y++)  
 {  
 if(nums.get(y)==x)  
 count++;  
 }  
  
 if(count>most)  
 most=x;  
 }  
  
 return most;  
 }  
}

Text

Description automatically generated

## Array of class

import java.util.\*;  
  
public class CountingVotes extends Votes{  
 static Scanner *sc* = new Scanner(System.*in*);  
 public static void main(String[] args) {  
 ArrayList<Votes> votes = *enterVotes*();  
 ArrayList<Double> percent = new ArrayList<>();  
 int total =0;  
  
 int numVotes =0;  
 Votes bestVote = new Votes();  
  
 //Find who had most votes  
 for(int x=0;x<5;x++)  
 {  
 if(votes.get(x).getVotes()>numVotes)  
 {  
 bestVote= votes.get(x);  
 numVotes=bestVote.getVotes();  
 }  
 }  
  
 //Find percent of votes  
 for(int x=0;x<5;x++)  
 {  
 total+=votes.get(x).getVotes();  
 }  
  
 double temp =0;  
 for(int x=0;x<5;x++)  
 {  
 temp =(votes.get(x).getVotes());  
 percent.add((temp/total)\*100);  
 }  
  
  
 //Print the Info  
 for(int x=0;x<5;x++)  
 {  
 System.*out*.println(votes.get(x).getCandidate() + " >>> " + votes.get(x).getVotes() +" votes >>> %"+ percent.get(x) + " of the votes");  
 }  
  
 System.*out*.println("\n" +bestVote.getCandidate() + " Had the most Votes");  
  
 }  
  
 public static ArrayList<Votes> enterVotes()  
 {  
 ArrayList<Votes> v = new ArrayList<>();  
 for(int x=0;x<5;x++)  
 {  
 System.*out*.println("Enter the name of Candidate " + (x+1));  
 String cand = *sc*.nextLine();  
  
 System.*out*.println("Enter the Votes recieved by Candidate "+ (x+1));  
 int votes = *sc*.nextInt();  
 *sc*.nextLine();  
  
 Votes c = new Votes(cand,votes);  
 v.add(c);  
 }  
  
 return v;  
 }  
}

Text

Description automatically generated

## GUI 2

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class GUI2 implements ActionListener {  
 private JLabel d1,d2,d3,gallons,cost;  
 private JPanel panel;  
 private JSpinner s1,s2,s3;  
 private JButton button;  
  
 GUI2() {  
  
 //Frame  
 JFrame frame = new JFrame("On Demand");  
 frame.setLayout(new FlowLayout());  
 frame.setSize(500,750);  
 frame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 frame.setVisible(true);  
 frame.setBounds(350,225,800,400);  
  
 //Labels  
 d1 = new JLabel("Length");  
 d2 = new JLabel("Width");  
 d3= new JLabel("Depth");  
 gallons = new JLabel("Gallons Needed >>> ");  
 cost = new JLabel("Cost >>> $");  
  
 //Spinners  
 s1 = new JSpinner();  
 s1.setPreferredSize(new Dimension(50,30));  
 s2 = new JSpinner();  
 s2.setPreferredSize(new Dimension(50,30));  
 s3 = new JSpinner();  
 s3.setPreferredSize(new Dimension(50,30));  
  
 //Button  
 button = new JButton("Calculate");  
 button.addActionListener(this);  
  
 //Panel  
 panel = new JPanel();  
 panel.setLayout(new FlowLayout(FlowLayout.*CENTER*));  
 panel.add(d1);  
 panel.add(s1);  
 panel.add(d2);  
 panel.add(s2);  
 panel.add(d3);  
 panel.add(s3);  
 panel.add(button);  
 panel.add(gallons);  
 panel.add(cost);  
 frame.add(panel);  
 }  
  
 public static void main(String[] args) {  
 new GUI2();  
 }  
  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 double g,c =0;  
  
 g=(int)(s1.getValue())\*(int)(s2.getValue())\*(int)(s3.getValue());  
  
 gallons.setText("Gallons Needed >>> "+ g);  
  
 c=g\*.0015;  
  
 cost.setText("Cost >>> $"+ c);  
 }  
}

Graphical user interface

Description automatically generated with medium confidence