What is your name? ______________________________________

Answer all questions to the best of your ability. An answer that is unreadable will be marked wrong.

You may **NOT** use a calculator on this test/exam.

If you need scratch paper, please ask. You may use margins and whitespace for your work and the back of this cover page.

Total Points = 100
Time = 80 min

I have not rendered or received unauthorized assistance in taking this examination. I will not discuss this exam with anyone in the course that has not yet taken it.

____

Signature

Part 1: Multiple Choice (3 X 5 = 15 pts)

Q1) What will be printed by the following code snippet, assuming that *`letters`* is an initially empty ArrayList?

```
letters.add("A");
letters.add("B");
letters.add("C");
letters.add(1, "D");
System.out.println(letters.toString());
```

A) 4
B) [D, A, B, C]
C) [A, B, C, D]
D) [A, D, B, C]

Q2) Consider the following code fragment:

```
for (int i = 0; i < 10; i++)
{
      for (int j = 0; j < 3; j++)
      {
            System.out.print("*");
      }
      System.out.println();
}
```

How many asterisk characters (*) will be printed when this code executes?

A) 10
B) 30
C) 3
D) 31

Q3) Which of the following will create a Run Time error?

A) Trying to read a double value from the keyboard using the Scanner method nextInt().
B) Typing system.out.println() for System.out.println().
C) Calculating a Box volume by the formula, length + width + depth.
D) A program that uses no variables.

Q4) What is the possible cause of this error?

```
WhichCar.java:44: possible loss of precision
   found    : double
   required: int
            cost = totalCost;
```

A) `cost` has not been declared
B) `cost` has not been initialized
C) `totalCost` has not been initialized
D) cost is an `int` and `totalCost` is a double

Q5) What is the value of *routeNumber* after the following statement is executed?

```
int zipCode = 93707, routeNumber;
switch (zipCode)
{
  case 93705:
  case 93706:
      routeNumber = 1;
      break;
  case 93710:
  case 93720:
      routeNumber = 2;
      break;
  default:
     routeNumber = 0;
     break;
}
```

A) 0
B) 1
C) 2
D) undefined

Part 2: Program output prediction (10 X 2 = 20 pts)

Q6) What will be the output of the following program?

```
int persons;
int age;
double tickets;
persons = 5;
age = 50;
if (persons == 1)
   if (age < 50)
      tickets = 12.50;
   else
      tickets = 10.00;
else if (persons > 1 && persons < 5)
   if (age <= 50)
      tickets = persons * 12.00;
   else
      tickets = persons * 9.50;
else
   tickets = persons * 9.00;
System.out.print(tickets);
```

Answer:

```
45.0
```

Q7) What will be the output of the following program?

```
class Output
{
    final static short i = 2;
    public static int j = 0;

    public static void main(String [] args)
    {
        for (int k = 0; k < 3; k++)
        {
            switch (k)
            {
                case i: System.out.print(" 0 ");
                case i-1: System.out.print(" 1 ");
                case i-2: System.out.print(" 2 ");
            }
        }
    }
}
```

Answer:

```
2  1  2  0  1  2
```

Some helpful Scanner for the upcoming problems:
`nextLine()`: returns an entire line up to and including the new line character
`next()`: returns the next token from a line
`nextInt()`: returns the next integer from a line
`nextDouble()`: returns the next double value from a line

Part 3: Evaluate the expression (10 pt)

Q8) If an expression cannot be evaluated, write **Invalid** in both the data type and result columns. Assume that variable *abc* has been declared to be a *double* and contains the value *12.5*.

Expression	Data type of result	Result
`(int)(6.0 / 2) == ((int)3.0 / 1.0)`	boolean	True
`(4 % 10) - (6 / 3.0)`	double	2.0
`abc.equals(10.0)`	Invalid	Invalid
`(“text” + 2 + 11).equals(“text13”) || true`	boolean	True
`0 <= 0 && !false`	boolean	True

Part 4: Trace command line arguments (10 pt)

Q9) Given the following Winter code:

```
public class Winter
{
   public static void main (String [] args)
   {
      int number;
      char letter;

      if (args.length > 1)
      {
         number = Integer.parseInt(args[0]);
         letter = args[1].charAt(0);
         for (int ii = number; ii >= 0; ii--)
         {
            System.out.print(letter);
         }
         System.out.println();
      }
      else
         System.out.println("Invalid input");
   }
}
```

a. What is output by the following command:

java Winter

```
Invalid Input
```

b. What is output by the following command:

java Winter 4 snow

```
sssss
```

Part 5: Solve a problem (15 pt)

Q10) Write a program that first gets a list of integers as input. The input begins with an integer indicating the number of integers that follow. Assume that the list will always contain fewer than 20 integers. That list is followed by two more integers representing lower and upper bounds of a range. Your program should output all integers from the list that are within that range (inclusive of the bounds). For coding simplicity, follow each output integer by a space, even the last one. The output ends with a newline.

Ex: If the input is:

```
5 25 51 0 200 33
0 50
```

then the output is:

```
25 0 33
```

(the bounds are 0-50, so 51 and 200 are out of range and thus not included in the output). **HINT:** To achieve the above, first read the list of integers into an array.

////// Your code starts here/////

```
import java.util.Scanner;

public class LabProgram {
   public static void main(String[] args) {
      Scanner scnr = new Scanner(System.in);
      int numValues;
      int lowerBound;
      int upperBound;
      int i;
      int userInput;
      int[] userNums = new int[20];

      numValues = scnr.nextInt();

      for (i = 0; i < numValues; ++i) {
         userInput = scnr.nextInt();
         userNums[i] = userInput;
      }

      lowerBound = scnr.nextInt();
      upperBound = scnr.nextInt();

      for (i = 0; i < numValues; ++i) {
         if (userNums[i] >= lowerBound && userNums[i] <= upperBound) {
            System.out.print(userNums[i] + " ");
         }
      }
      System.out.println();

   }
}
```

Part 6: Solve a problem (30 pt)

Q11) Complete the following methods

```
/**
 * a class to represents class grades
 *
 * @author - YOUR NAME
 * @version - 1.0 - DATE
 */
public class FinalExamGrade
{
   private double coding; // coding grade
   private double written; // written grade

   /** Default constructor, sets the two attributes to 0
     */
   public FinalExamGrade()
   {
      this.coding = 0;
      this.written = 0;
   }

   /** Explicit value constructor. This constructor
     *  should set the two grades.
     *
     *  If either of the two values is less than 0,
     *  set both values to 0
     *
     *  @param coding The score for the coding part
     */
   public FinalExamGrade(double coding, double written)
   {

      if (coding > 0 && written > 0)
      {
         this.coding = coding;
         this.written = written;
      }
      else
      {
         this.coding = 0;
         this.written = 0;
      }

   }

   /** Calculates the grade based on the two scores
     *  If the sum of the scores is between 90 - 100 (inc) the grade is
     *  'A'. If 80 - 89.9999 (not including 90), 'B', 70-79.9999 'C', 60-69.9999 'D'
     *  and less than 60 'F'.
     *
     *  @return The corresponding exam grade
     **********************************************************/
   public char calcGrade()
   {
      char grade;

      if (this.written + this.coding >= 90)
         grade = 'A';
      else if (this.written + this.coding >= 80)
         grade = 'B';
      else if (this.written + this.coding >= 70)
         grade = 'C';
      else if (this.written + this.coding >= 60)
         grade = 'D';
      else
         grade = 'F';

      return grade;

   }

   /** update will reset this object's written and coding scores based on
     *  the values being passed in.
     *  If either score is less than 0 do not change the scores.
     *
     * @param coding The new coding score
     * @param written The new written score
     */
   public void update(double coding, double written)
   {
      if (coding > 0 && written > 0)
      {
         this.coding = coding;
         this.written = written;
      }
   }

   /**
    * equals determines if two FinalExam objects are the same.
    *
    * The two objects are considered the same if the sum of the
    * two individual scores is the same. For example, if one had
    * coding: 20 and written: 50, and the next had coding: 15 and written: 55,
    * they would be consider the same (both total 70).
    *
    * @param other The FinalExam object to compare to this FinalExam object.
    * @return true if the this FinalExam is the same as other,
false otherwise.
     */
    public boolean equals (FinalExam other)
    {
       boolean result;
       result = false;

       if (this.coding + this.written == other.coding + other.written)
          result = true;

       return result;

    }

    /** makeCopy creates a new FinalExam object that is identical in values
      * to this Grade object
      *
      *  @return  A new FinalExam with the same values as the this object.
      */
    public FinalExam makeCopy()
    {

       return new FinalExam(this.coding, this.written);

    }


    /** countGrade
      *
      *  This method returns the count of the letters that match
      *  the score's calcGrade value.
      *  If the array is null or empty, return -1.
      *  For example, if the array were {'A', 'B', 'C', 'A'}
      *  and the FinalExamGrade values were coding: 25, written: 50
      *  this method would return 1.
      *  However, using the same array, if the FinalExamGrade values were
      *  coding 20 and written 45 this method would return 0.
      *
      *  @param arr1 The array of char values
      *  @param score The FinalExamGrade object to compare
      *  @return The count of the grade matches in the array
      */
    public static int countGrade(char [] arr1, FinalExamGrade score)
    {
       int result = -1;
       char grade = score.calcGrade();

       if (arr1 != null && arr1.length > 0)
       {
          result = 0;

          for (int ind = 0; ind < arr1.length; ind++)
             if (grade == arr1[ind])
                result++;
       }

       return result;

    }
 }
```

-----EXAM ENDS HERE-----