CSci 1103
Final

Name: __________________________________________

Student ID: ____________________________

Instructions: Please pick and answer any 10 of the 12 problems for a total of 100 points. If you answer more than 10 problems, only the first 10 will be graded. The time limit is 120 minutes. Please write your answers in the space provided. The exam is open book and notes. You may use electronic devices to ONLY look at either an e-book version or electronic notes. You may not use the internet, compiler or any other outside resources. (If you are typing on your keyboard/input device for anything other than ctrl-F to find words in the e-book or notes, this is probably not acceptable.)
**Problem (1)** [10 points] Write (in Java) a `getSlope()` method for the class below. A slope can be found by this formula: \( \frac{y_2 - y_1}{x_2 - x_1} \). Simply write the definition of `getSlope()`, you do not need to declare it.

```java
public class Problem1 {
    private double startX;
    private double endX;
    private double startY;
    private double endY;

    public double getSlope()
    {
        return (endY - startY)/(endX - startX);
    }
}
```
Problem (2) [10 points] Write (in Java) a evenArray() method, which returns an array of even numbers (starting from 0 going up) of the size specified by the input. For example, for the code segment below the array length should be 10 (containing numbers 0 through 18).

```java
int[] evenNumbers = evenArray(10);

public static int[] evenArray(int sz)
{
    int[] result = new int[sz];
    for(int i=0; i < sz; i++)
    {
        result[i] = i*2;
    }

    return result;
}
```
Problem (3) [10 points] Make a J-Frame which has a single button you can click. If this button is pressed, print “Beep!” to the terminal.

```java
public class Problem3 implements ActionListener{

    public static void main(String[] args) {
        JFrame win = new JFrame();
        JButton happyHappyJoyJoy = new JButton();
        happyHappyJoyJoy.addActionListener(new Problem3());
        win.add(happyHappyJoyJoy);
        //win.setSize(100, 100); // optional
        win.setVisible(true); // tiny window, but okay...
    }

    @Override
    public void actionPerformed(ActionEvent ae) {
        System.out.println("Beep! ");
    }
}
```
Problem (4) [10 points] Write (in Java) a “Height” class to make the main() code below work.

```java
public static void main(String[] args) {
    Height h = new Height(5, 10); // h is 5'10" (5 foot, 10 inches)
    System.out.println(h); // prints: 5'10"
}
```

```java
public class Height {

    private int ft;
    private int in;

    public Height(int f, int i)
    {
        ft = f;
        in = i;
    }

    @Override
    public String toString()
    {
        return ft + "\" + in + "\"
    }

}
```
Problem (5) [10 points] Consider the “Point” class below. Write (in Java) a constructor for this class that takes in an array and initializes x to the first element in this array and y to the second. (You can assume the array has at least two elements.)

```java
public class Point {
    private double x;
    private double y;
}

public Point(double[] nums) {
    x = nums[0];
    y = nums[1];
}
```
Problem (6) [10 points] Write (in Java) a midPoint() method (usable anywhere) that takes as input two Point variables (as defined in problem 5). Assume you are making this method in the Point class. This function should correctly find and return the midpoint of the two Points passed in as input (the formula for this case is: $x_{\text{midpoint}} = \frac{x_1+x_2}{2}$, $y_{\text{midpoint}} = \frac{y_1+y_2}{2}$). An example code of its use is given below:

```java
public static void main(String[] args) {
    Point p1;
    Point p2;
    // magically initialize p1 and p2 to something
    Point mid = midPoint(p1, p2);
}

public static Point midPoint(Point a, Point b) {
    Point result = new Point();
    result.x = (a.x+b.x)/2;
    result.y = (a.y+b.y)/2;
    return result;
}

... or using problem 5...

public static Point midPoint(Point a, Point b) {
    double[] vals = new double[2];
    vals[0] = (a.x+b.x)/2;
    vals[1] = (a.y+b.y)/2;
    Point result = new Point(vals);
    return result;
}
```
Problem (7) [10 points] Assume that there exists a file “numbers.txt” that contains multiple integers. Write a Java code segment that displays the range of numbers in this file (range = max - min).

```java
try {
    Scanner fin = new Scanner(new FileInputStream("numbers.txt"));

    double max = Double.NEGATIVE_INFINITY;
    double min = Double.POSITIVE_INFINITY;
    while(fin.hasNextDouble())
    {
        double value = fin.nextDouble();
        if(value > max)
        {
            max = value;
        }
        if(value < min)
        {
            min = value;
        }
    }

    System.out.println("Range = " + (max-min));
} catch (FileNotFoundException ex) {
    // could just put Exception ^^<-- there
    System.out.println("Houston... we have a problem...");
}
```
Problem (8) [10 points] Find 3 possible places for errors in the following code. Assume this is all the code except for imports. Explain specifically what causes each error and whether it is a syntax or logic error:

```java
public class Problem8 {
    private int num;

    public Problem8() {
        2 = num;
    }

    public getValue() {
        return num;
    }

    public static void main(String[] args) {
        Problem8 x;
        System.out.println(getValue());
    }
}
```

Lines:
2 = num;
Syntax, cannot store things into "2".

Lines:
    public getValue() {
Syntax, no return type... Only constructors don't have a return type.

Lines:
    System.out.println(getValue());
Syntax, getValue() is not static, so would have to say x.getValue().

Lines:
private int num;
Syntax, Either this needs to be a "private static int", so it always has a value, or you need to initialize "x" down in main() so the constructor runs:
    Problem8 x = new Problem8();
Problem (9) [10 points] Consider the Java code below. Describe 3 different ways in which the catch() statement can be run (i.e. "oops" will be displayed). Also indicate whether this is a checked or unchecked exception. You do not need to say the exact name of the Exception, but just what could cause it to happen.

```java
public static void main(String[] args) {
    try{
        Scanner in = new Scanner(System.in);
        System.out.println("What file do you want to write to?");
        String fileName = in.next();

        PrintWriter fout = new PrintWriter(new FileOutputStream(fileName));

        System.out.println("What index to write?");
        int index = in.nextInt();
        fout.println(fileName.charAt(index));
    }
    catch(Exception e)
    {
        System.out.println("oops");
    }
}
```

1. Entering just ".", as this is not a valid file name (or any invalid file name)
   IOException
   Checked
2. Second input as "bannana pants", not an int...
   InputMismatchException
   Unchecked
3. Second input as "-1", is not valid for charAt(-1)
   ArrayIndexOutOfBoundsException
   Unchecked
Problem (10) [10 points] Write a segment of code in Java (i.e. pretend you are writing somewhere in main()) that creates a 10 by 10 array. The first row and column (i.e. top row and left column) should be 1. Every other cell in the array should be the sum of the cell to the left plus the cell above itself.

```java
int[][] x = new int[10][10];
for(int i=0; i < 10; i++)
{
    for(int j=0; j < 10; j++)
    {
        if(i==0 || j==0)
        {
            x[i][j] = 1;
        }
        else
        {
            x[i][j] = x[i-1][j] + x[i][j-1];
        }
    }
}
... or ...

int[][] x = new int[10][10];
for(int i=0; i < x.length; i++)
{
    x[i][0] = 1;
    x[0][i] = 1;
}

for(int i=1; i < x.length; i++)
{
    for(int j=1; j < x[i].length; j++)
    {
        x[i][j] = x[i-1][j] + x[i][j-1];
    }
}
```
Problem (11) [10 points] Consider the main() method below. Write code that will make this main() syntactically correct (i.e. you can run it) and “x.foo()” and “y.foo()” prints out two different things.

```java
public static void main(String[] args) {
    a x = new a();
    a y = new b();
    x.foo();
    y.foo();
}
```

```java
public class a{
    public void foo() {
        System.out.println("I AM _-*A*_");
    }
}
```

```java
public class b extends a{
    // @Override // optional
    public void foo() {
        System.out.println("I AM _-*B*_");
    }
}
```
Problem (12) [10 points] Write (in Java) a single if-statement that is true on the set of integers shown and false for all values not shown. A “...” signifies that the pattern continues in the specified direction (i.e. a “...” on the left indicates the pattern continues for smaller numbers). An example is provided below.

```
int i: 0, 1, 2, 3, 4, ...
```

Answer:
```
if(i >= 0)
  or
if(i > -1)
```

(a) int i: ... -10, 0, 10, 20, 30, 40, ...

(b) int i: ... -2, 2, 4, 6, 8, 12, 14, 16, 18, 22, ...
   (look carefully)

(c) int i: 0, 1, 2, 3, 4, 6, 7, 8, 9, 10
   (look carefully)

(d) int i: ... -8, -6, -4, -2, 12, 14, 16, 18, 20, ...
    (you might want parenthesis for this)

(e) int i: 1, 2, 3, 4, 5, 10, 11, 12, 13, 14, 15
    (parenthesis might be necessary here too)

(a) if(i%10==0)
(b) if(i%2==0 && i%10!=0)
(c) if(i >= 0 && i <= 10 && i != 5)
(d) if(i%2==0 && (i < 0 || i > 10))
(e) if((i > 0 && i < 6) || (i > 9 && i < 16))