Fill in this information:

Name (clearly print last, first, middle):

Net ID: _____

CU ID: _____

I have followed the rules of academic integrity on this exam (sign):

Circle your lab section (just <u>one</u> location!):

We distribute CS100M graded exams in *lab section*. Uncollected exams will be available in Carpenter B101 during consulting hours. We will post in **Announcements** when the exams are ready at Carpenter.

	12:20	1:25	2:30	3:35
Tue	Upson B7	Red Room	Plue Poom	Plue Poom
	Blue Room	Blue Room	Blue Room	Blue Room
Wed	Upson B7	Red Room		Blue Room
	Blue Room	Blue Room	Dide Room	Dide Room

Instructions:

Failure to follow any instruction may result in a point deduction on your exam:

- Turn off all cell phones, beepers, pagers, and any other devices that will interrupt the exam.
- Remove all calculators, reference sheets, or any other material. This test is closed book.
- Fill out the information at the top of this exam.
- Wait for us to announce when the exam begins.
- Skim all problems first! Read each problem completely before starting it.
- Write your solutions directly on the test using blue/black pen or pencil. Clearly indicate which problem you are solving. You may write on the back of each sheet. If you need scrap paper, ask a proctor.
- Provide only one statement, expression, value, or comment per blank!
- Do not alter, add, or remove any code that surrounds blanks and boxes.
- Do not supply multiple answers. If you do so, we will choose which one to grade.
- Follow good style! When possible, keep solutions general, avoid redundant code, use descriptive variables, use named constants, indent substructures, avoid breaking out of loops, use information hiding, and maintain other tenets of programming philosophy.
- · Comment each control structure, major variable, and function (if used), briefly.
- You have 91 minutes. Budget about one minute per point.
- Do not work on bonus problems (if there are any) until you have thoroughly proofread all required (core-point) problems!
- Try to figure out problems yourself before raising your hand so that we can avoid disturbing people in cramped rooms.

Core Points:



Initial or Name:

Problem 1 [10 points] General Concepts

Answer the following questions. Be concise and clear.

la [1 point] What is $0.346 \div 0.3$? Give a numerical result that is accurate to at least two decimal places. If you get stuck, show your work.

Ib [1 point] Let *i* be the imaginary number $i = \sqrt{-1}$, x = 2 + 3i, and y = 1 - 2i. What is xy?

Ic [1 point] *OOP* stands for _____

1d [1 point] As opposed to MATLAB, Java is ______ typed.

le [2 points] Explain the principle of *type promotion*. Give a brief example.

If [2 points] In terms of a programming language, what is *scope*?

lg [2 points] Distinguish between a *class* and *object*.

Problem 2 [15 points] Java Language

Fill in the boxes for the following Java code fragments. We have typed this page exactly as we have meant! Consider each problem as a separate Java code fragment. If the code will produce an error or exception during compilation or execution, write **error** as the output.

problem	code fragment	output
2a [1 point]	System.out.println("hi!"); // we've given the answer	hi!
2b [1 point]	System.out.println(9/5);	
<i>2c</i> [1 point]	<pre>int x = 1; boolean b = (boolean) x; System.out.println(b);</pre>	
2d [1 point]	<pre>int x = 4; int y = x; x = 2; System.out.println(y);</pre>	
<i>2e</i> [1 point]	<pre>System.out.println(""+1+2+3);</pre>	
<i>2f</i> [1 point]	<pre>int x = 1; System.out.println(1+(x++)); System.out.println(x);</pre>	
<i>2g</i> [2 points]	<pre>int x = 2; if (x = 3) System.out.println(x);</pre>	
<i>2h</i> [3 points]	<pre>int sum = 0; int x; for (x = 4 ; x > -2 ; x) sum += x; System.out.println(x+sum);</pre>	
<i>2i</i> [3 points]	<pre>public class Thing { public static boolean d; public static void main(String[] args) { int a; { char b; { int c; } } char c; System.out.println(d); } }</pre>	
<i>2j</i> [1 point]	<pre>class _ { \$_; _() { System.out.println(_\$_); } } public class { public static void main(String[] _) { new _(); } }</pre>	

Problem 3 [35 points] Control structures

Problem: Suppose that a device generates a sequence of a random number (1 to 40, inclusive) of random bits. Each bit (0 or 1) has an equal probability of being generated. Complete program **Problem3** that simulates the device by generating a sequence of bits and reporting the following information:

- The entire sequence of bits from left to right.
- The largest number of successively generated 1s.

Sample Session: Suppose the simulation generated a sequence of eight bits in this order, from left to right: 0, 0, 1, 1, 1, 1, 0, 1. For this example, the program would print this report:

```
Sequence: 00111101
Maximum sequence of 1s: 4
```

Specifications:

- Do not use strings, arrays, or objects in your solution, which will be solely contained in one Main Class (Problem3).
- Do not count or report the maximum number of 0s.
- You must use fields **MAXBITS**, **sequence**, and **maxOnes**. Do not write or use additional fields.
- You will need to complete methods **randBit**, **randInt**, and **runSim**, which are specified below. Remember that **Math.random()** returns a random *double* between 0 and 1, including only 0. Do not write methods that have not already been defined in this problem.

```
public class Problem3 {
   public static final int MAXBITS = randInt(1,40); // # of bits in sim
   public static String sequence = "";
                                                     // seq of bits generated in sim
   public static int maxOnes;
                                                     // max number of 1s in sim
   // Run program:
   public static void main(String[] args) {
      runSim(); // run one simulation to set sequence and maxOnes
      System.out.println("Sequence: " + sequence );
      System.out.println("Maximum sequence of 1s: " + maxOnes);
   }
   // Return a random integer (LOW <= integer <= HIGH):</pre>
   public static int randInt(int low, int high) {
      if (low > high) System.exit(0);
      return _
                                                                                     ;
   }
   // Return a random bit (0 or 1) using method randInt:
   public static int randBit() {
      return
                                                                                     ;
   }
```

```
// Problem3 continues on next page
```

// Run // and public	the simulation, which updates the sequence of bits (sequence) determines the max sequence of 1s (maxOnes): static void runSim() {
- int	oneCounter =; // count of 1s so far
11	Generate and process each bit from count = 1 to MAXBITS:
for	(;;) {

} // end for

} // Method main

} // Class Problem3

Problem 4 [40 points] OOP

Problem: You need to complete two classes **Triangle** and **Point** that driver class **Problem4** uses to compare the position of **Triangles** that are supplied as points p1, p2, and p3 in terms of (x, y) coordinates. As shown in the following figure, **Triangle** A is to the left of **Triangle** B. Assume that the **Triangles**' bases will always be aligned with the x axis and that the vertices are specified in this order: p1, p2, p3. Assume p2 is always to the left of p1.



Use principles of good object-oriented programming (especially information hiding for fields!) when using and/or completing the following classes:

- **Problem4**: Method **main** creates two **Triangles**, outputs the area of the **Triangle** created first, and determines which **Triangle** is leftmost. We have written the complete code for you.
- **Point**: This class represents a vertex coordinate (**x**, **y**).
- Triangle: This class represents a triangle with three coordinates (p1, p2, p3), which are objects of class Point.
 - Method **leftmost** compares the current **Triangle** with another **Triangle** by comparing the leftmost horizontal coordinates of each **Triangle**. Return either **Triangle** if neither is leftmost.
 - Method **area** computes the area of the current **Triangle** using this formula: $area = (base \times height)/2$.
 - Other methods: You need to complete the toString method and the constructor. We have provided accessors.

Example session:

```
Area: 2.0
Triangle: A
```

```
public class Problem4 {
   public static void main(String[] args) {
      Triangle t1 = new Triangle("A", new Point(1,2), new Point(0,0), new Point(2,0));
      Triangle t2 = new Triangle("B", new Point(2,3), new Point(1,0), new Point(4,0));
      System.out.println("Area: "+t1.area());
      System.out.println(t2.leftmost(t1));
   }
} // Class Problem4
class Point {
                                       x; // horizontal coord
                                      _ y; // vertical coord
   // Create a new Point:
      public Point(double a, double b) { x = a; y = b; }
   // Accessors:
      public double getX() { return x; }
      public double getY() { return y; }
} // Class Point
```

CS100M: Prelim 3	Initial or Name:	Page 7
class Triangle {		
	name; // name of Triangle	
	pl; // (x,y) of pl (top co	oord)
	p2; // (x,y) of p2 (left,	bottom coord)
	p3; // (x,y) of p3 (right)	, bottom coord)
<pre>// Construct a new Triangle calle public Triangle(String name, F</pre>	ed NAME from coords P1, P2, P3: Point p1, Point p2, Point p3) { = name ;	

```
_____ = p1;
        _____ = p2;
                _____ = p3;
}
// Accessors for coordinates:
  public Point getP1() { return p1; }
  public Point getP2() { return p2; }
  public Point getP3() { return p3; }
// Compare current Triangle with supplied Triangle (other) and
// return leftmost Triangle:
  public Triangle leftmost(Triangle other) {
```

```
// Compute and return the area of the current Triangle:
    public double area() {
                = Math.abs( _____);
      double base
      double height = Math.abs( ______);
      return
                                                             _ ;
  }
  // Stringify the current Triangle:
    public String toString() {
      return
                                                             _ ;
  }
} // Class Triangle
```

Bonus: [5 points] What is the output for this perfectly legal Java program?

```
class Friend {
  private String Friend;
  private Friend friend;
  public Friend(String Friend) { this.Friend = Friend; }
  public void Friend(Friend friend) { this.friend = friend; friend.friend = this; }
  public String friend() { return friend.Friend; }
}
public class Friends {
  public static void main(String[] friends) {
      { Friend Friend = new Friend("friend.friend"); }
     Friend Friend = new Friend("Friend");
     Friend friend = new Friend("friend");
     Friend.Friend(friend);
      System.out.println(friend.friend());
   }
}
// Output:
```