CSCI 136 Written Exam #2 **Name: ___________________________________**
Fundamentals of Computer Science II
Spring 2012

This exam consists of 6 problems on the following 8 pages.

You may use your double-sided hand-written 8 ½ x 11 note sheet during the exam. No computers, mobile devices, cell phones, or other communication devices of any kind are permitted.

If you have a question, raise your hand and I will stop by. Since partial credit is possible, **please write legibly and show your work**.

Problem	**Points**	**Score**
1	**8**	
2	**10**	
3	**10**	
4	**14**	
5	**10**	
6	**12**	
Total	**64**	

1. **Strings** (8 points). Consider the following program:

```
public class Prob1
{
   public static void main(String [] args)
   {
      String a = args[0].toLowerCase();
      String b = args[1].toLowerCase();
      for (int i = 0; i < Math.min(a.length(), b.length()); i++)
      {
         if (a.charAt(i) == b.charAt(i))
            System.out.print(a.charAt(i));
         else
            System.out.print("-");
      }
   }
}
```

Below are four example executions of the program. Give the output produced by the program. If the given input would cause a runtime error, write "runtime error".

Command line	Output
`% java Prob1 BOB`	
`% java Prob1 BOB robert`	
`% java Prob1 tom ----- tommy`	
`% java Prob1 aBcD AbCd`	

2. **Methods** (10 points). Match the method description on the left with the ***best*** answer on the right. Not every letter will be used. Each letter will be used ***at most once***.

____ A class that `implements ActionListener` must have this method.

____ A class that `implements Runnable` must have this method.

____ Your GUI has a class `MyPanel` that extends `JPanel`. This method is called when `repaint()` is executed on the `JFrame` containing your `MyPanel` object.

____ When you run a *.class file using the `java` program, this method is the one that must exist and immediately gets executed.

____ A class that `implements Serializable` must have this method.

A. `public void run(Thread t)`

B. `public void runnable(String name)`

C. `public void run()`

D. `public void actionPerformed(ActionEvent e)`

E. `public void actionPerformed()`

F. `public void buttonClicked(ActionEvent e)`

G. `public static void main()`

H. `private static void main(String [] args)`

I. `public void main(String [] args)`

J. `public static void main(String [] args)`

K. `public void init(Graphics g)`

L. `public void paintComponent(Graphics g)`

M. `public void repaint(Graphics g)`

N. Trick question, this interface does not require any particular method be implemented.

O. Trick question, doing this would result in a runtime error.

P. Trick question, no method is executed in this situation.

3. **Methods** (10 points). Match the method description on the left with the ***best*** method on the right. Assume candidate methods appear inside a class named BigInt. Not every letter will be used. Each letter will be used ***at most once***.

_____ A method that can return the count of BigInt objects in an array that are equal to a specified BigInt object.

_____ You have an object c of type BigInt. This method is called implicitly by the following line:

```
  System.out.println(c);
```

_____ A method that can return whether one BigInt is less than, equal to, or greater than another BigInt.

_____ A constructor that creates a BigInt based on another BigInt.

_____ A method that can return a new array of a specified number of BigInt objects.

```
A. public boolean foo(BigInt [] a)
B. public int foo(BigInt [] a, BigInt t)
C. public BigInt[] foo(int t)
D. public String toString()
E. public static String toString()
F. public void toString(String s)
G. public int foo(BigInt a)
H. public boolean foo(BigInt a)
I. public enum foo(BigInt a, BigInt b)
J. public void foo(BigInt a)
K. public BigInt()
L. public BigInt(BigInt a)
M. public static BigInt(BigInt a)
N. private BigInt(BigInt a)
```

4. **Multiple choice** (2 points each). For each question, circle the ***ONE*** best answer.

a) Which of the following Java expressions does ***NOT*** always correctly compute the mathematical average of the integer variables a, b, c and d? Assume a, b, c, and d are integers in the range [-1000, 1000].

```
  I. (double) ((a + b + c + d) / 4.0);
 II. ((double) (a + b + c + d)) / 4.0;
III. (a + b + c + d) / 4;
 IV. (a + b + c + d) / 4.0;
  V. (a + (double) b + c + d) / 4;
```

b) Consider the following recursive method:

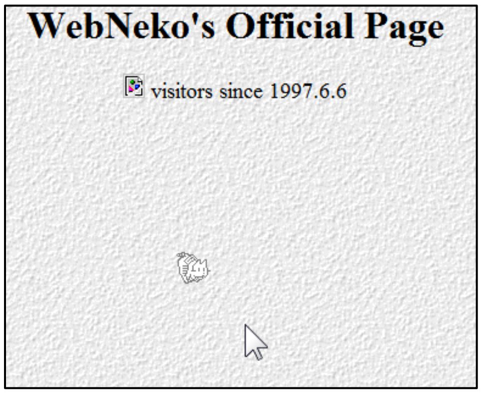
```
public static int foo(int n)
{
   if (n == 4)
      return 2;
   else
      return 2 * foo(n + 1);
}
```

What is the value returned by the method call `foo(2)`?

```
  I. 2
 II. 4
III. 8
 IV. 16
  V. 24
```

c) You would like to include an element on your web page that displays an animated cat. The cat attempts to chase the mouse cursor whenever the cursor is inside the element's section on the web page. Which of the following technologies could ***best*** be used to achieve this?

I. Java web start
II. Java applet
III. Java JFrame application
IV. Java REST web service
V. Java object serialization



d) All the following are true ***EXCEPT***:

I. When compiling to native code, the final executable will only run on a specific hardware architecture.
II. The bytecode of a Java program compiled on Mac OS will run in Windows 7 or Linux without needing the original source code.
III. A disadvantage of bytecode approaches such as Java and .NET is that compile-time errors can only be detected when the program is run in the virtual machine.
IV. A just-in-time (JIT) compiler improves performance by caching frequently needed conversions from bytecode to native instructions.

e) When writing an object to disk using Java object serialization, what keyword do you use to specify that an instance variable should ***NOT*** be written to disk?

I. static
II. private
III. transient
IV. synchronized
V. final

f) Which of the following is ***true*** about accessing a REST web service via Java:

I. Communication with the remote web service can be done via the HTTP protocol using Java's `URL` and `URLConnection` classes.
II. The first step is to download the JAR file containing the *.class file
III. The first step is to download the XML code contained in the *.jnlp file
IV. The remote web server must have Java installed
V. The remote web server must be in the client's security sandbox

g) Consider the following short program:

```java
int x = 1;
int y = -4;
while ((x >= 0) || (x <= y))
{
   x--;
   y++;
}
System.out.println("x = " + x + ", y = " + y);
```

What values of x and y are printed after the `while` loop halts?

I. x = 1, y = -4
II. x = 0, y = -3
III. x = -1, y = -2
IV. x = -2, y = -1
V. The loop never halts

5. **OOP** (10 points). Assume you have the following Java classes (some details have been omitted):

```
public abstract class Supply
{
   private double cost;
   public double getCost() { return cost; }
   public abstract String toString();
}
public abstract class Writing extends Supply
{
   protected boolean erasable;
   public boolean getErasable() { return erasable; }
   public void write(String s)  { /* stuff */      }
}
public class Coffee extends Supply
{
   private double volume;
   public void drink(int amount) { volume -= amount; }
   public String toString()      { return "coffee!"; }
}
public class Pencil extends Writing { /* stuff */ }
public class Pen extends Writing    { /* stuff */ }
```

a) What method(s) must be implemented in the classes `Pencil` and `Pen` in order for them to compile?

b) Which of the above classes could you successfully instantiate an object of?

c) Which of the above classes could you instantiate and add to the following variable named `list`:

```
ArrayList<Writing> list = new ArrayList<Writing>();
```

d) What inherited method(s) could you potentially *override* in the class Pen? Circle the correct method(s).

```
void drink(int amount)              public double getCost()

public boolean getErasable()        public void write(String s)
```

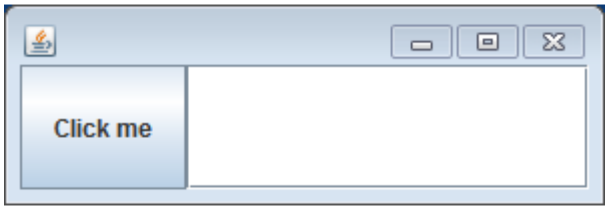
e) In an instance method in the `Pen` class, what inherited instance variables could you access directly (that is without using a getter/setter method)?

6. **GUIs** (12 points). Consider the following GUI program:

```
public class Click extends JFrame implements ActionListener
{
   private JTextField field;
   public Click()
   {
      JButton button = new JButton("Click me");
      add(button, BorderLayout.WEST);
      button.addActionListener(this);

      field = new JTextField();
      field.addActionListener(new Handler());
      add(field, BorderLayout.CENTER);

      setSize(300, 100);
      setVisible(true);
   }
   public void actionPerformed(ActionEvent e)
   {
      field.setText("" + Integer.parseInt(field.getText()) / 2);
   }
   class Handler implements ActionListener
   {
      public void actionPerformed(ActionEvent e)
      {
         field.setText("" + Integer.parseInt(field.getText()) * 2);
      }
   }
   public static void main(String [] args) { Click click = new Click(); }
}
```



a) When the application starts, it appears as shown above. What happens if the user immediately clicks the "Click me" button? Explain why and describe how you might prevent this.

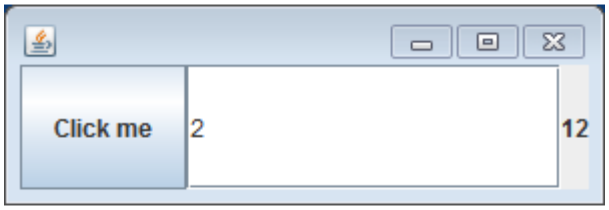
b) Assume the user does the following: types 7 into the text field, hits enter, and then clicks the "Click me" button. What result is displayed in the text field?

c) Assume the user has typed 5 into the text field. Give the result shown in the text field after four consecutive clicks of the "Click me" button.

Before any click	1st click	2nd click	3rd click	4th click
5				

6. **GUI** (continued)

d) Add code to the previous GUI so that a text label appears to the right of the text field (e.g. as shown in the screenshot below). The label should start at a value of 0 and increment by one every time the button is pressed. We have reproduced the previous program below. Add or change the code below as necessary.



```
public class Click extends JFrame implements ActionListener
{
   private JTextField field;


   public Click()
   {
      JButton button = new JButton("Click me");
      add(button, BorderLayout.WEST);
      button.addActionListener(this);

      field = new JTextField();
      field.addActionListener(new Handler());
      add(field, BorderLayout.CENTER);




      setSize(300, 100);
      setVisible(true);
   }




   public void actionPerformed(ActionEvent e)
   {
      field.setText("" + Integer.parseInt(field.getText()) / 2);



   }

   class Handler implements ActionListener
   {
      public void actionPerformed(ActionEvent e)
      {
         field.setText("" + Integer.parseInt(field.getText()) * 2);



      }
   }
   public static void main(String [] args)
   {
      Click click = new Click();


   }
}
```