CSE 11
Midterm
Fall 2011

Page 1 _________ (18 points)
Page 2 _________ (18 points)
Page 3 _________ (31 points)
Page 4 _________ (13 points)
Page 5 _________ (8 points)
Total _________ (88 points = 84 base points + 4 points EC [5%])

(84 points = 100%)

This exam is to be taken by yourself with closed books, closed notes, no electronic devices.
You are allowed one side of an 8.5"x11" sheet of paper handwritten by you.
### (Partial) Operator Precedence Table

<table>
<thead>
<tr>
<th>Operators</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>! ++ --</td>
<td>(pre &amp; post inc/dec) right to left</td>
</tr>
<tr>
<td>* / %</td>
<td>left to right</td>
</tr>
<tr>
<td>+ -</td>
<td>left to right</td>
</tr>
<tr>
<td>&lt; &lt;= &gt; &gt;=</td>
<td>left to right</td>
</tr>
<tr>
<td>== !=</td>
<td>left to right</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>=</td>
<td>right to left</td>
</tr>
</tbody>
</table>

1) What are the values of the indicated variables after the following code segments are executed?

```java
int x = 5, y = 3, z;
boolean bool1 = !((x > 4) || (y <= 6)) == ((y <= 4) && !(x > 6));
if ( x++ >= 4 || --y >= 3 )
    z = x++ + --y;
else
    z = ++x + y--;
```

```java
int a = 5, b = 3, c;
boolean bool2 = !(b > 4) && (a <= 6) && (a <= 4) || (b > 6);
if ( a++ >= 4 && --b >= 3 )
    c = a++ + --b;
else
    c = ++a + b--;
```

What gets printed?

```java
public class While
{
    public static void main( String[] args )
    {
        final int MAX = 9, MIN = 5;
        int i = 7, j = 8;

        while ( i <= MAX )
        {
            while ( j > MIN )
            {
                ++j;
                System.out.println( i + " " + j );
                j -= 4;
            }
            i++;
            j = i;
        }
        System.out.println( i + " " + j );
    }
}
```
2) Which of the following are valid Java identifiers? (Circle your answer(s).)

- 1stJavaClass
- My-First-Java-Class
- sEvEnTeEn
- CSE11Is_1
- CSE11Is#1
- CSE_11
- My1stJavaClass
- float

Given the following definition of class Thing2, what is the output of the Java application Test2?

class Thing2 {
    private int count;
    public Thing2( int count ) {
        this.count = count;
    }
    public int getCount() {
        return this.count;
    }
    public void setCount( int count ) {
        this.count = count;
    }
    public String toString() {
        String s = " ";
        switch( this.count ) {
            case 3: s = s + "tres "; break;
            case 2: s = s + "duo "; break;
            case 1: s = s + "uno "; break;
            default: s = s + "mucho "; break;
        }
        return s;
    }
    public void swap1( Thing2 t2 ) {
        Thing2 temp;
        Thing2 t1 = this;
        temp = t1;
        t1 = t2;
        t2 = temp;
    }
    public void swap2( Thing2 t2 ) {
        int temp;
        temp = this.getCount();
        this.setCount( t2.getCount() );
        t2.setCount( temp );
    }
}

public class Test2 {
    public static void main( String[] args ) {
        Thing2 first = new Thing2( 4 );
        Thing2 second = new Thing2( 2 );
        Thing2 temp = first;
        first = second;
        second = temp;
        System.out.println( first.toString() );
        System.out.println( second.toString() );
        Thing2 third = new Thing2( 1 );
        Thing2 fourth = new Thing2( 3 );
        third.swap2( fourth );
        System.out.println( third.toString() );
        System.out.println( fourth.toString() );
        first.setCount( third.getCount() );
        fourth = second;
        System.out.println( first == third );
        System.out.println( second == fourth );
        System.out.println( first.toString().equals( third.toString() ) );
        System.out.println( second.toString().equals( fourth.toString() ) );
        System.out.println( first.toString() );
        System.out.println( second.toString() );
        System.out.println( third.toString() );
        System.out.println( fourth.toString() );
        first.swap1( second );
        System.out.println( first.toString() );
        System.out.println( second.toString() );
    }
}

**Output**

---

---

---

---

---

---

---

---

---

---

---

---
3) What output is produced by the following program?

```java
1 public class Test3
2 {
3   private int a;
4   private static int b = 2;
5   private int c;
6
7   public static void main(String[] args)
8   {
9     Test3 ref = new Test3(3);
10     ref.method1(ref.a);
11   }
12
13   public Test3(int a)
14   {
15     this.a = a;
16   }
17
18   public void method1(int x)
19   {
20     int c = x--;
21     int b;
22
23     b = a + 2;
24     a = c + 3;
25
26     System.out.println("this.a = " + this.a);
27     System.out.println("Test3.b = " + Test3.b);
28     System.out.println("this.c = " + this.c);
29     System.out.println("b = " + b);
30     System.out.println("c = " + c);
31     System.out.println("result = " + method2(b + c));
32     System.out.println("this.a = " + this.a);
33     System.out.println("Test3.b = " + Test3.b);
34     System.out.println("this.c = " + this.c);
35     System.out.println("x = " + x);
36     System.out.println("a = " + a);
37     System.out.println("b = " + b);
38     System.out.println("c = " + c);
39   }
40
41   private int method2(int x)
42   {
43     int b = x;
44     int c = this.c + Test3.b;
45
46     x = a = b + c;
47
48     System.out.println("this.a = " + this.a);
49     System.out.println("Test3.b = " + Test3.b);
50     System.out.println("this.c = " + this.c);
51     System.out.println("x = " + x);
52     System.out.println("a = " + a);
53     System.out.println("b = " + b);
54     System.out.println("c = " + c);
55
56     Test3.b = b + 2;
57     this.c = a + c;
58
59     return x + 5;
60   }
61  }
```

```
Output
this.a = ________
Test3.b = ________
this.c = ________
   c = ________
      b = ________
a = ________
   result = ________
this.a = ________
Test3.b = ________
this.c = ________
   x = ________
a = ________
   b = ________
   c = ________
```
4) What is the output of this recursive method if it is invoked as `ref.mystery(10)`? Draw Stack Frames to help you answer this question.

```java
public int mystery(int a)
{
    int b = a - 2;
    if (b >= 7)
    {
        System.out.println(a + " " + b);
        a = b - mystery(b + 1);
    }
    else
    {
        System.out.println("Stop");
        b = a + 2;
    }
    System.out.println(a + " " + b);
    return a + b;
}
```

What gets printed if the value of the actual argument passed to this method is 0?

```java
public void f5(int x)
{
    int y = 0;
    if (x <= 1)
        y = 3;
    else if (x <= 2)
        y = 5;
    else if (x == 3 || x >= 4)
        y = 7;
    else
        y = 9;
    System.out.println(y);
}
```

What gets printed if the value of the actual argument passed to this method is 0?

```java
public void f5(int x)
{
    int y = 0;
    if (x <= 1)
        y = 3;
    else if (x <= 2)
        y = 5;
    else if (x == 3 || x >= 4)
        y = 7;
    else
        y = 9;
    System.out.println(y);
}
```

What is the output of this recursive method if it is invoked as `ref.mystery(10)`? Draw Stack Frames to help you answer this question.

```java
int mystery(int a)
{
    int b = a - 2;
    if (b >= 7)
    {
        System.out.println(a + " " + b);
        a = b - mystery(b + 1);
    }
    else
    {
        System.out.println("Stop");
        b = a + 2;
    }
    System.out.println(a + " " + b);
    return a + b;
}
```
5) Given the following definitions:

```java
public interface Doable {
    public abstract String doit();
}
```

```java
public class Thing1 implements Doable {
    private String str;
    public Thing1() {
        this.str = "Me";
    }
    public String speak() {
        return this.str;
    }
    public String doit() {
        return "Thing1 did it!";
    }
}
```

```java
public class Thing2 implements Doable {
    private String str;
    public Thing2() {
        this.str = "No, Me";
    }
    public String speak(String s) {
        return this.str + s;
    }
    public String doit() {
        return "Thing2 does it too!";
    }
}
```

And the following variable definitions:

```java
Thing1 thing1 = new Thing1();
Thing2 thing2 = new Thing2();
Doable doable;
```

What gets printed with the following statements (each statement is executed in the order it appears). If there is a compile time error, write "Error".

```java
doable = thing1;
System.out.println( doable.getClass().getName() );
System.out.println( doable.doit() );
System.out.println( thing1.speak() );
doable = thing2;
System.out.println( doable.getClass().getName() );
System.out.println( doable.doit() );
System.out.println( thing2.speak( " Here" ) );
```

What two changes/additions would be needed to the above interface and class definitions so `doable.speak()` would compile and run for all valid assignments to `doable`? Be specific what needs to be added to which file(s). Do not remove or change any of the existing code.

1) 

2)
Scratch Paper