CSE 11
Midterm
Fall 2013

Page 1 _________ (20 points)
Page 2 _________ (9 points)
Page 3 _________ (32 points)
Page 4 _________ (15 points)
Page 5 _________ (13 points)
Page 6 _________ (21 points)

Total _________ (110 points = 105 base points + 5 points EC [~5%])
(105 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>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 a = 6, b = 2, d;
boolean c = !(b > 6) && (a >= 3) && (a <= 4) || (b < 6);
if ( a++ >= 4 && --b >= 2 )
    d = ++a + b--;
else
    d = a++ + --b;
```

```java
int x = 6, y = 2, w;
boolean z = !((x > 4) || (y <= 6)) == ((y <= 4) && (x > 6));
if ( x++ >= 4 || --y >= 3 )
    w = --x + y++;
else
    w = x-- + ++y;
```

What gets printed?

```java
public class While
{
    public static void main( String[] args )
    {
        final int MAX = 11, MIN = 4;
        int i = 6, j = 8;

        while ( i <= MAX )
        {
            while ( j > MIN )
            {
                j -= 2;
                System.out.println( i + " " + j );
            }
            i += 3;
            j = i;
        }

        System.out.println( i + " " + j );
    }
}
2) What gets printed?

```java
int a = 2;
int b = 4;
int c = 6;
System.out.println( a + b + (c + " = ") + a + (b + c) );
```

What is the output produced by the following program? (Hint: Draw stack frames)

```java
public class Swap {
    private int a;

    public Swap(int a) {
        this.a = a;
    }

    public void swap(int a, int b) {
        int tmp;
        tmp = a;
        a = b;
        b = tmp;
    }

    public void swap(Swap ref) {
        int tmp;
        tmp = this.a;
        this.a = ref.a;
        ref.a = tmp;
    }

    public static void swap(Swap ref1, Swap ref2) {
        Swap tmp;
        tmp = ref1;
        ref1 = ref2;
        ref2 = tmp;
    }

    public static void main(String[] args) {
        int a = 44; Swap ref1;
        int b = 11; Swap ref2;
        ref1 = new Swap(3);
        ref2 = new Swap(7);
        Swap.swap(ref1, ref2);
        System.out.println(ref1.a);
        System.out.println(ref2.a);

        ref1 = new Swap(3);
        ref2 = new Swap(7);
        ref1.swap(a, b);
        System.out.println(a);
        System.out.println(b);

        ref1 = new Swap(3);
        ref2 = new Swap(7);
        ref1.swap(ref2);
        System.out.println(ref1.a);
        System.out.println(ref2.a);
    }
}
```

The different swap() method definitions have the same name but differ in their formal parameters. This is an example of method ________________.
3) What output is produced by the following program?

```java
public class Test3 {
    private static int a;
    private int b;
    private int c = 3;

    public static void main(String[] args) {
        Test3 ref = new Test3(5);
        ref.method1(Test3.a);
    }

    public Test3(int b) {
        this.b = b;
    }

    private void method1(int x) {
        int c = x + 4;
        int b;
        b = a + 2;
        a = c + 3;

        System.out.println("Test3.a = " + Test3.a);
        System.out.println("this.b = " + this.b);
        System.out.println("this.c = " + this.c);
        System.out.println("c = " + c);
        System.out.println("b = " + b);
        System.out.println("a = " + a);
        System.out.println("x = " + x);
        System.out.println("result = " + method2(11));
    }

    public int method2(int x) {
        int a = x;
        int c = b;
        x = b;

        System.out.println("Test3.a = " + Test3.a);
        System.out.println("this.b = " + this.b);
        System.out.println("this.c = " + this.c);
        System.out.println("c = " + c);
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("x = " + x);
        System.out.println("result = " + method2(11));
    }

    public int method2(int x) {
        int a = x;
        int c = b;
        x = b;

        System.out.println("Test3.a = " + Test3.a);
        System.out.println("this.b = " + this.b);
        System.out.println("this.c = " + this.c);
        System.out.println("c = " + c);
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("x = " + x);
        System.out.println("result = " + method2(11));
    }
}
```

Output

```
Test3.a = ________
this.b = ________
this.c = ________
c = ________
b = ________
a = ________
x = ________
result = ________
Test3.a = ________
this.b = ________
this.c = ________
x = ________
a = ________
b = ________
c = ________
```
What is the output of this recursive method if it is invoked as `ref.mystery( 5 );`? Draw Stack Frames to help you answer this question.

```
int mystery( int a )
{
    int b = a + 3;
    if ( b <= 11 )
    {
        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 by the following code? _______
```
int x = 13;
if ( x > 7 )
{
    x += 3;  // Same as x = x + 3;
}
if ( x >= 15 )
{
    x += 4;
}
System.out.println( x );
```

What gets printed by the following code? _______
```
int x = 13;
if ( x < 7 )
{
    x += 3;  // Same as x = x + 3;
}
if ( x >= 10 )
{
    x += 4;
}
System.out.println( x );
```

What gets printed by the following code? _______
```
int x = 13;
if ( x > 7 )
{
    x += 3;  // Same as x = x + 3;
}
if ( x <= 12 )
{
    x += 4;
}
System.out.println( x );
```

What gets printed by the following code? _______
```
int x = 13;
if ( x < 7 )
{
    x += 3;  // Same as x = x + 3;
}
if ( x >= 15 )
{
    x += 4;
}
System.out.println( x );
```

What gets printed by the following code? _______
```
int x = 13;
if ( x > 7 )
{
    x += 3;  // Same as x = x + 3;
}
if ( x >= 15 )
{
    x += 4;
}
System.out.println( x );
```

What gets printed by the following code? _______
```
int x = 13;
if ( x > 7 )
{
    x += 3;  // Same as x = x + 3;
}
if ( x <= 12 )
{
    x += 4;
}
System.out.println( x );
```

What gets printed by the following code? _______
```
int x = 13;
if ( x < 7 )
{
    x += 3;  // Same as x = x + 3;
}
if ( x >= 10 )
{
    x += 4;
}
System.out.println( x );
```
5) Given the following definitions:

```
5) Given the following definitions:

36
5

And the following variable definitions in another class:

Thing1 thing1 = new Thing1();
Thing2 thing2 = new Thing2();
Printable printable;

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".

```

<table>
<thead>
<tr>
<th>Statement</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>printable = thing1;</code></td>
<td></td>
</tr>
<tr>
<td><code>System.out.println( printable.print( true ) );</code></td>
<td></td>
</tr>
<tr>
<td><code>System.out.println( thing1.print() );</code></td>
<td></td>
</tr>
<tr>
<td><code>System.out.println( printable.print() );</code></td>
<td></td>
</tr>
<tr>
<td><code>printable = thing2;</code></td>
<td></td>
</tr>
<tr>
<td><code>System.out.println( printable.print( &quot;CS11FZZ&quot; ) );</code></td>
<td></td>
</tr>
<tr>
<td><code>System.out.println( printable.print( false ) );</code></td>
<td></td>
</tr>
<tr>
<td><code>System.out.println( thing2.print( &quot;CS11FZZ&quot; ) );</code></td>
<td></td>
</tr>
</tbody>
</table>

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

1)

2)
6) Trace the following program and specify its output.

```java
public class Trace {
    public static void main( String[] args ) {
        System.out.println( "main1" );
        foo3();
        System.out.println( "main2" );
        foo2();
        System.out.println( "main3" );
        foo1();
    }

    public static void foo1() {
        System.out.println( "A" );
    }

    public static void foo2() {
        System.out.println( "B" );
        foo1();
        System.out.println( "C" );
    }

    public static void foo3() {
        System.out.println( "D" );
        foo2();
        System.out.println( "E" );
    }
}
```

What is the equivalent Java expression for the following expression such that no ! operators are used? ( != is a different operator than ! )

```java
!( x >= 42 || y != 37 )
```

What is the default initial value of an instance variable that is defined as a boolean? 

What is the default initial value of an instance variable that is defined as an object reference? 

What is the default initial value of an instance variable that is defined as an int? 

What is the default initial value of a local variable that is defined as a double? 

---

If `b` is a boolean variable, then the statement

```java
b = ( b == false );
```

has what effect? _____

A) It causes a compile-time error message.
B) It causes a run-time error message.
C) It causes `b` to have the value false regardless of its value just before the statement was executed.
D) It always changes the value of `b`.
E) It changes the value of `b` if and only if `b` had value true just before the statement was executed.

Which of the following is equivalent to and has the same effect as

```java
b = ( b == false );
```

? _____

A) `b = ( b == true );`
B) `b = ( b != true );`
C) `b = ( b != false );`
D) `b = ( b == b );`
E) `b = ( b != b );`
F) More than one of the above statements is equivalent