CSE 11
Midterm
Fall 2009

Page 1 ___________ (12 points)
Page 2 ___________ (24 points)
Page 3 ___________ (30 points)
Page 4 ___________ (23 points)
Page 5 ___________ (12 points)

Total ___________ (101 points = 96 base points + 5 points EC [5%])
### 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 $x$ and $y$ (left) and $a$ and $b$ (right) after the following code segments are executed?

```java
int x = 2, y = 4;
if ( x++ >= 3 || --y >= 3 )
    x = x++ + --y;
else
    x = ++x + y--;
```

```java
int a = 2, b = 4;
if ( a++ >= 3 && --b >= 3 )
    a = a++ + --b;
else
    a = ++a + b--;
```

Assume we have a Java source file named Program.java and it uses at least one class in the objectdraw library. Write the full Unix command to compile this Java program.

```
This command will produce a file named:
```

Write the full Unix command to run this as a Java application.

```
Assume we have correctly written a Program.html file. Write the full Unix command to run the above program as an applet.

```
```
2) Given the following definition of class Thing1, what is the output of the Java application Question2?

```java
public class Thing1 {
    private int count;

    public Thing1( int count ) {
        this.count = count;
    }

    public int getCount() {
        return this.count;
    }

    public void setCount( int count ) {
        this.count = count;
    }

    public String toString() {
        if ( this.count == 1 )
            return "one";
        else if ( this.count == 2 )
            return "two";
        else if ( this.count == 3 )
            return "three";
        else
            return "too many";
    }
}

public class Question2 {
    public static void main( String[] args ) {
        Thing1 first = new Thing1( 3 );
        Thing1 second = new Thing1( 4 );
        System.out.println( first.toString() );
        System.out.println( second.toString() );
        Thing1.swap1( first, second );
        System.out.println( first.toString() );
        System.out.println( second.toString() );
        Thing1 third = new Thing1( 1 );
        Thing1 fourth = new Thing1( 2 );
        second.setCount( third.getCount() );
        first = fourth;
        System.out.println( first.toString() );
        System.out.println( second.toString() );
        System.out.println( third.toString() );
        System.out.println( fourth.toString() );
        System.out.println( first.toString().equals( fourth.toString() ) );
        System.out.println( second.toString().equals( third.toString() ) );
        System.out.println( first == fourth );
        System.out.println( second == third );
    }
}
```

Output

```
 too many
 two
 one
 three
```

3) What output is produced by the following program?

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

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

    public Test3()
    {
        c = 3;
    }

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

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

    private int method2( int x )
    {
        int a = x;
        int c = this.c + Test3.a;

        x = b = a + c;

        System.out.println( "Test3.a = " + Test3.a );
        System.out.println( "this.b = " + this.b );
        System.out.println( "this.c = " + this.c );
        System.out.println( "a = " + a );
        System.out.println( "b = " + b );
        System.out.println( "c = " + c );

        Test3.a = a + 2;
        this.b = b + c;

        return x + 3;
    }
}
```

Output

Test3.a = 
this.b = 
this.c = 
c = 
b = 
a = 

Test3.a = 
this.b = 
this.c = 
a = 
b = 
c = 
result = 

Test3.a = 
this.b = 
this.c = 
a = 
b = 
c = 
x = 

Use the numbers below to identify various program parts.

1) static method      2) constructor
3) class definition (type) 4) instance method
5) static variable      6) local variable
7) instance variable   8) formal parameter
9) actual argument

_____ Test3() on line 11     _____ a on line 38
_____ method2() on line 36   _____ c on line 5
_____ Test3 on line 1        _____ a on line 3
_____ ref.c on line 9        _____ x on line 15
_____ main() on line 6       _____ ref on line 8
4) What gets printed in the following code fragment?

```java
final int MAX = 6;
int i = 3;
int j;

while ( ++i < MAX )
{
  j = 12;
  while ( j > MAX + i )
  {
    System.out.println( i + " " + j);
    j--;
  }
  System.out.println( i + " " + j);
}
```

What is the output of this recursive method if it is invoked as `ref.mystery( 6 );`? Draw Stack Frames to help you answer this question.

```java
int mystery( int a )
{
  int b = a + 3;
  if ( b > 5 )
  {
    System.out.println( a + " " + b );
    a = b + mystery( a - 2 );
    System.out.println( a + " " + b );
  }
  else
  {
    System.out.println( "Cease" );
    System.out.println( a + " " + b );
    b = a - 3;
    System.out.println( a + " " + b );
  }
  return a + b;
}
```
5) Given the following definitions:

```java
public interface Speakable
{
    public String speak();
}
```

```java
public class Puppy implements Speakable
{
    private static final String
    PUPPY_SPEAK = "Bark";

    public Puppy()
    {
        // ctor initialization here
    }

    public String speak()
    {
        return PUPPY_SPEAK;
    }

    public String wag()
    {
        return "wag wag";
    }
}
```

```java
public class Kitty implements Speakable
{
    private static final String
    KITTY_SPEAK = "Meow";

    public Kitty()
    {
        // ctor initialization here
    }

    public String speak()
    {
        return KITTY_SPEAK;
    }

    public String sleep( int time )
    {
        return time + " second cat nap";
    }
}
```

And the following variable definitions:

```java
private Puppy puppy;
private Kitty kitty;
private Speakable speakable;
```

Indicate what gets printed with the following statements (each statement is executed in the order it appears).

```java
puppy = new Puppy();
kitty = new Kitty();
speakable = kitty;
System.out.println( speakable.getClass().getName() ); ____________________________
System.out.println( speakable.speak() );   ____________________________
```

```java
speakable = puppy;
System.out.println( speakable.getClass().getName() ); ____________________________
System.out.println( speakable.speak() );   ____________________________
```

```java
System.out.println( puppy.speak() );   ____________________________
System.out.println( puppy.wag() );    ____________________________
System.out.println( kitty.speak() );   ____________________________
System.out.println( kitty.sleep( 1000 ) );   ____________________________
```

What two things would we need to change in Speakable.java, Puppy.java, and/or Kitty.java in order to have Kitty and Puppy objects listen for and handle ActionEvents? Be specific what needs to change in which file(s).