Signature _____ **Name _____**

cs11f ____ **Student ID _____**

CSE 11
Midterm
Fall 2010

Page 1 _____ (16 points)

Page 2 _____ (14 points)

Page 3 _____ (30 points)

Page 4 _____ (16 points)

Page 5 _____ (10 points)

Total _____ (86 points = 82 base points + 4 points EC [5%])

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

Operators				Associativity
!	++	-- (pre & post inc/dec)		right to left
*	/	%		left to right
+	-			left to right
<	<=	>	>=	left to right
==	!=			left to right
&&				left to right
||				left to right
=				right to left

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

```
int x = 3, y = 5;
boolean z = !((x > 4) || (y <= 6)) == ((y <= 4) && (x > 6));

if ( x++ >= 4 || --y >= 3 )
  x = x++ + --y;
else
  x = ++x + y--;
```

x =
y =
z =

```
int a = 3, b = 5;
boolean c = !(b > 4) && (a <= 6) && (a <= 4) || (b > 6);

if ( a++ >= 4 && --b >= 3 )
  a = a++ + --b;
else
  a = ++a + b--;
```

a =
b =
c =

What gets printed?

```
public class Loops
{
   public static void main( String[] args )
   {
      final int MAX = 6, MIN = 4;
      int i = -2, j = -3;

      for (i = MAX; i > MIN; --i )
      {
         j = 5;
         while ( j <= MAX )
         {
            System.out.println( i + " " + j );
            ++j;
         }
      }

      System.out.println( i + " " + j );

   } // end main()
}
```

2) Given the following definition of class Thing2, what is the output of the Java application Question2?

```
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()
  {
    if ( this.count == 5 )
      return "five";
    else if ( this.count == 6 )
      return "six";
    else if ( this.count == 7 )
      return "seven";
    else
      return "need more";
  }

  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 Question2
{
  public static void main( String[] args )
  {
    Thing2 first = new Thing2( 5 );
    Thing2 second = new Thing2( 4 );

    Thing2 temp = first;
    first = second;
    second = temp;

    System.out.println( first.toString() );
    System.out.println( second.toString() );

    Thing2 third = new Thing2( 7 );
    Thing2 fourth = new Thing2( 6 );;

    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 = new Thing2( 4 );
    second = new Thing2( 7 );

    first.swap1( second );

    System.out.println( first.toString() );
    System.out.println( second.toString() );
  }
}
```

Output

3) What output is produced by the following program?

```
 1  public class Test3
 2  {
 3    private static int a;
 4    private int b;
 5    private int c;

 6    public static void main( String[] args )
 7    {
 8      Test3 ref = new Test3( 5 );

 9      ref.method1( ref.c );
10    }

11    public Test3( int c )
12    {
13      this.c = c;
14    }

15    public void method1( int x )
16    {
17      int c = ++x;
18      int b;

19      b = c + 3;
20      a = b + 2;

21      System.out.println( "Test3.a = " + Test3.a );
22      System.out.println( "this.b = " + this.b );
23      System.out.println( "this.c = " + this.c );
24      System.out.println( "c = " + c );
25      System.out.println( "b = " + b );
26      System.out.println( "a = " + a );
27      System.out.println( "result = " + method2( c + b ) );
28      System.out.println( "Test3.a = " + Test3.a );
29      System.out.println( "this.b = " + this.b );
30      System.out.println( "this.c = " + this.c );
31      System.out.println( "a = " + a );
32      System.out.println( "b = " + b );
33      System.out.println( "c = " + c );
34      System.out.println( "x = " + x );
35    }

36    private int method2( int x )
37    {
38      int a = x;
39      int c = this.c + Test3.a;

40      x = b = a + c;

41      System.out.println( "Test3.a = " + Test3.a );
42      System.out.println( "this.b = " + this.b );
43      System.out.println( "this.c = " + this.c );
44      System.out.println( "a = " + a );
45      System.out.println( "b = " + b );
46      System.out.println( "c = " + c );

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

49      return x + 3;
50    }
51  }
```

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 letters below to identify various program parts.

A) class definition (type)
B) local variable
C) static method
D) instance variable
E) actual argument
F) instance method
G) static variable
H) constructor
I) formal parameter

_____ `Test3()` on line 11

_____ `method2()` on line 36

_____ `Test3` on line 1

_____ `ref.c` on line 9

_____ `main()` on line 6

_____ `a` on line 38

_____ `c` on line 5

_____ `a` on line 3

_____ `x` on line 15

_____ `ref` on line 8

4) What gets printed as a result of the call `Q4( 3, -1 )`? _____

```
public void Q4( int a, int b )
{
  if ( (a > 0) && (b > 0) )
  {
    if ( a > b )
    {
      System.out.println( "A" );
    }
    else
    {
      System.out.println( "B" );
    }
  }
  else if ( (a < 0) || (b < 0) )
  {
    System.out.println( "C" );
  }
  else
  {
    System.out.println( "D" );
  }
}
```

Give an example of values passed as arguments to Q4() that would result in the method printing "D".

`Q4( _____ , _____ );`

What is the output of this recursive method if it is invoked as `ref.mystery( 9 );`? Draw Stack Frames to help you answer this question.

```
int mystery( int a )
{
  int b = a + 3;

  if ( b > 6 )
  {
    System.out.println( a + " " + b );
    a = b + mystery( a - 3 );
    System.out.println( a + " " + b );
  }
  else
  {
    System.out.println( a + " " + b );
    b = a - 4;
    System.out.println( a + " " + b );
    System.out.println( "Whoa" );
  }

  return a + b;
}
```

Output

5) Given the following definitions:

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

```
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";
  }
}
```

```
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:

```
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). If there is a compile time error, write "Error".

```
puppy = new Puppy();
kitty = new Kitty();

speakable = puppy;

System.out.println( speakable.getClass().getName() );   _____

System.out.println( speakable.wag() );                  _____

System.out.println( speakable.speak() );                _____

System.out.println( puppy.wag() );                      _____

speakable = kitty;

System.out.println( speakable.getClass().getName() );   _____

System.out.println( kitty.wag() );                      _____

System.out.println( speakable.speak() );                _____

System.out.println( speakable.sleep( 1000 ) );          _____
```

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

1)
2)

Scratch Paper