Signature _____ **Name _____**

cs11f ____ **Student ID _____**

CSE 11
Midterm
Fall 2008

Page 1 _____ (10 points)

Page 2 _____ (22 points)

Page 3 _____ (23 points)

Page 4 _____ (17 points)

Page 5 _____ (12 points)

Total _____ (84 points = 80 base points + 4 points EC [5%])

(Partial) Operator Precedence Table

Operators				Associativity
*	/	%		left to right
+	-			left to right
<	<=	>	>=	left to right
==	!=			left to right
&&				left to right
||				left to right
=				right to left

1) Which of the following <u>are not</u> valid Java identifiers? (Circle your answer(s).)

```
1stAndTen        double_down      First&Ten        main

choice?          Upper-Case       five5five        _420_
```

2) Using the operator precedence table above, evaluate each expression and state what gets printed. Remember short-circuit evaluation with && and ||.

```
int x = 6;
int y = -2;
int z = 13;
boolean b = !(x - 6 < y || z == 2 * x + 1);

System.out.print( "b = " + b );        _____________________

b = (x + y < z || 4 * y + z > x) && x > y;

System.out.print( "b = " + b );        _____________________

x = z + x % 4 + y * 2;

System.out.print( "x = " + x );        _____________________
```

3) What gets printed with each of the following statements?

```
int a = 1;
int b = 3;
int c = 5;
System.out.println( a + (b + c) + " = " + (a + b) + c );
```

_____

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

_____

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

_____

4) Write a single method that draws a filled circle on the canvas when the mouse is clicked. The circle should be 100 by 100 pixels centered at the point of the mouse click. Here is the signature for the FilledOval constructor:

```
FilledOval( double x, double y, double width, double height, DrawingCanvas canvas)
```

5) Assume we have a Java source file named Tunes.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 Tunes.html file. Write the full Unix command to run the above program as an applet.

_____

6) What gets printed in the following program fragment?

```
final int MAX = 4;
int i = 2;
int j;

while ( i++ < MAX )
{
  j = 7;

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

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

7) What output is produced by the following program?

```
 1 public class Test7
 2 {
 3   private int a;
 4   private boolean b;
 5   private static int c = 42;

 6   public static void main( String[] args )
 7   {
 8     Test7 ref = new Test7();

 9     ref.method1( 5 );
10   }

11   public Test7()
12   {
13     a = 1;
14   }

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

19     b = this.a + 2;
20     this.a = b * 3;

21     System.out.println( "this.a = " + this.a );
22     System.out.println( "this.b = " + this.b );
23     System.out.println( "c = " + c );
24     System.out.println( "b = " + b );
25     System.out.println( "a = " + a );
26     System.out.println( "method2() result = " + method2( x + b ) );
27     System.out.println( "this.a = " + this.a );
28     System.out.println( "this.b = " + this.b );
29   }

30   private int method2( int x )
31   {
32     int a = x;
33     int c = this.a + a;

34     b = a != c;

35     System.out.println( "a = " + a );
36     System.out.println( "b = " + b );
37     System.out.println( "c = " + c );
38     System.out.println( "this.a = " + this.a );
39     System.out.println( "this.b = " + this.b );

40     this.a = a + 2;
41     this.b = b == false;

42     return x + 3;
43   }
44 }
```

Output:

```
this.a = __________

this.b = __________

c = __________

b = __________

a = __________

a = __________

b = __________

c = __________

this.a = __________

this.b = __________

method2() result = __________

this.a = __________

this.b = __________
```

Use the numbers below to identify various program parts.

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

_____ `Test7` on line 1

_____ `b` on line 4

_____ `main()` on line 6

_____ `Test7()` on line 11

_____ `method1()` on line 15

_____ `a` on line 3

_____ `x` on line 15

_____ `ref` on line 8

_____ `a` on line 17

_____ `c` on line 5

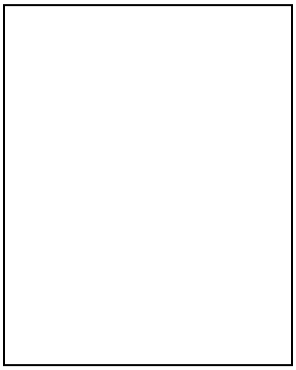
8) Given the following if – else if sequence below, fill in the blanks to produce an equivalent result with a switch statement.

```
int x = /* some value */;
String str;

if ( x / 2 == 2 )
  str = "2 stars";
else if ( x / 2 == 4 )
  str = "4 stars";
else if ( x / 2 == 6 )
  str = "6 stars";
else
  str = "a comet";

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

```
switch( _______________ )
{
  _______________:
    str = "2 stars";

    _______________;

  _______________:
    str = "4 stars";

    _______________;

  _______________:
    str = "6 stars";

    _______________;

  _______________:
    str = "a comet";

    _______________;
}

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

9) What is the output of this recursive method if it is invoked with the actual argument of 5, as in `ref.mystery( 5 );` ? Draw Stack Frames to help you answer this question.

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

  if ( b > 5 )
  {
    a = mystery( a - 1 );
    System.out.println( a + " " + b );
  }
  else
  {
    a = b + 3;
    System.out.println( a + " " + b );
  }

  return b;
}
```



10) 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).

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

System.out.println( puppy.speak() );                    _____________________________

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

System.out.println( kitty.speak() );                    _____________________________

System.out.println( kitty.sleep( 2000 ) );              _____________________________

speakable = puppy;

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

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

speakable = kitty;

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

System.out.println( speakable.speak() );                _____________________________
```

What two things would we need to change in Speakable.java, Puppy.java, and/or Kitty.java in order for the statement `speakable.sleep( 1000 );` to compile and work properly? Be specific.

1)

2)

Scratch Paper