Signature ______________________ **Name _________________________**

cs11f ____ **Student ID _____________________**

CSE 11
Midterm
Fall 2012

Page 1 ___________ (20 points)

Page 2 ___________ (17 points)

Page 3 ___________ (31 points)

Page 4 ___________ (15 points)

Page 5 ___________ (8 points)

Page 6 ___________ (20 points)

Total ___________ (111 points = 105 base points + 6 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

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 = 4, y = 6, z;
boolean bool1 = !((x > 4) || (y <= 6)) == ((y <= 4) && !(x > 6));

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

bool1 =

x =

y =

z =

```
int a = 4, b = 6, c;
boolean bool2 = !(b > 4) && (a <= 6) && (a <= 4) || (b > 6);

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

bool2 =

a =

b =

c =

What gets printed?

```
public class While
{
  public static void main( String[] args )
  {
    final int MAX = 10, MIN = 5;
    int i = 7, j = 7;

    while ( i < MAX )
    {
      while ( j >= MIN )
      {
        ++j;
        System.out.println( i + " " + j );
        j -= 4;  // j = j - 4;
      }
      i++;
      j = i;
    }

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

2) What is printed by the following code?

```
int foo = 37;
int bar = 42;
boolean foobar = ( foo == bar );
System.out.println( foobar );             ____________
bar = 37;
System.out.println( foobar );             ____________

System.out.println( foo == bar );         ____________
```

Given the following definition of class Thing1, what is the output of the Java application Test2?

```
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()
  {
    String s = " ";

    switch( this.count )
    {
      case 1:
        s = s + "1st ";
        break;

      case 2:
        s = s + "2nd ";
        break;

      case 3:
        s = s + "3rd ";

      default:
        s = s + "rest ";
        break;
    }

    return s;
  }

  public void swap1( Thing1 t1 )
  {
    Thing1 temp;
    Thing1 t2 = this;

    temp = t1;
    t1 = t2;
    t2 = temp;
  }

  public void swap2( Thing1 t1 )
  {
    int temp;

    temp = this.getCount();
    this.setCount( t1.getCount() );
    t1.setCount( temp );
  }
}
```

```
public class Test2
{
  public static void main( String[] args )
  {
    Thing1 first = new Thing1( 1 );
    Thing1 second = new Thing1( 2 );

    first.swap1( second );

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

    Thing1 third = new Thing1( 3 );
    Thing1 fourth = new Thing1( 4 );;

    Thing1 temp = third;
    third = fourth;
    fourth = temp;

    System.out.println( third.toString() );
    System.out.println( fourth.toString() );

    third = first;
    fourth.setCount( second.getCount() );

    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 Thing1( 1 );
    second = new Thing1( 3 );

    first.swap2( 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 int a;
 4    private static int b = 2;
 5    private int c;

 6    public static void main( String[] args )
 7    {
 8      Test3 ref = new Test3( 4 );

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

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

15    private void method1( int x )
16    {
17      int c = x--;
18      int b;

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

21      System.out.println( "this.a = " + this.a );
22      System.out.println( "Test3.b = " + Test3.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( b + c ) );
28      System.out.println( "this.a = " + this.a );
29      System.out.println( "Test3.b = " + Test3.b );
30      System.out.println( "this.c = " + this.c );
31      System.out.println( "x = " + x );
32      System.out.println( "a = " + a );
33      System.out.println( "b = " + b );
34      System.out.println( "c = " + c );
35    }

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

40      x = a = b + c;

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

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

50      return x + 5;
51    }
52  }
```

Use the numbers below to identify various program parts.

1) local variable
2) instance variable
3) static method
4) class definition (type)
5) actual argument

6) static variable
7) formal parameter
8) constructor
9) instance method

_____ `main()` on line 6

_____ `Test3` on line 1

_____ `method2()` on line 36

_____ `Test3()` on line 11

_____ `ref.a` on line 9

_____ `x` on line 40

_____ `b` on line 4

_____ `c` on line 39

_____ `c` on line 5

_____ `c` on line 11

Output

```
this.a = ________
Test3.b = ________
this.c = ________
c = ________
b = ________
a = ________
this.a = ________
Test3.b = ________
this.c = ________
x = ________
a = ________
b = ________
c = ________
result = ________
this.a = ________
Test3.b = ________
this.c = ________
x = ________
a = ________
b = ________
c = ________
```

4)

What gets printed by the following code? ________

```
int x = 12;
if ( x > 7 )
{
  x += 3;  // Same as x = x + 3;
}
else
{
  x += 6;
}
System.out.println( x );
```

What gets printed by the following code? ________

```
int x = 12;
if ( x < 7 )
{
  x += 3;  // Same as x = x + 3;
}
else if ( x <= 10 )
{
  x += 6;
}
System.out.println( x );
```

What gets printed by the following code? ________

```
int x = 12;
if ( x < 7 )
{
  x += 3;  // Same as x = x + 3;
}
else
{
  x += 6;
}
System.out.println( x );
```

What gets printed by the following code? ________

```
int x = 12;
if ( x > 7 )
{
  x += 2;  // Same as x = x + 2;
}
else if ( x >= 10 )
{
  x += 6;
}
System.out.println( x );
```

What is the output of this recursive method if it is invoked as `ref.mystery( 8 );`? Draw Stack Frames to help you answer this question.

```
int mystery( int a )
{
  int b = a + 2;

  if ( b <= 11 )
  {
    System.out.println( a + " " + b );
    a = b + mystery( b - 1 );
  }
  else
  {
    System.out.println( "Whoa" );
    b = a - 2;
  }

  System.out.println( a + " " + b );
  return a - b;
}
```

Output

5) Given the following definitions:

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

```
public class Thing1 implements Speakable
{
  private String str;

  public Thing1()
  {
    this.str = "Thing1";
  }

  public String speak()
  {
    return this.str;
  }

  public String doit()
  {
    return "Thing1 did it!";
  }
}
```

```
public class Thing2 implements Speakable
{
  private String str;

  public Thing2()
  {
    this.str = "Thing2";
  }

  public String speak()
  {
    return this.str;
  }

  public String doit( String s )
  {
    return "Thing2 " + s;
  }
}
```

And the following variable definitions:

```
Thing1 thing1 = new Thing1();
Thing2 thing2 = new Thing2();
Speakable speakable;
```

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".

```
speakable = thing1;

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

System.out.println( speakable.doit() );           ____________________________

System.out.println( thing1.doit( "Here" ) );      ____________________________

speakable = thing2;

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

System.out.println( speakable.doit() );           ____________________________

System.out.println( thing2.doit() );              ____________________________
```

What two changes/additions would be needed to the above interface and class definitions so `speakable.doit( "Do it" )` would compile and run for all valid assignments to `speakable`? 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.

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

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

  public static void foo2()
  {
    System.out.println( "B" );
    foo3();
    System.out.println( "C" );
  }

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

}
```

What is the default initial value of a local variable that is defined as an int? _________________

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 a double? ____________

Will the following code compile? ___________
If not, what change do you need to make to the method header (not the method body) so that it will compile? Explain. Be specific.

```
public boolean test( int x )
{
  System.out.println( "In test" );
  return x * x;
}
```

Scratch Paper