**Signature** _____ **Name** _____

**cs11f** ____ **Student ID** _____

# 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**

| 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 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;
```

```
a =
b =
c =
d =
```

```
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;
```

```
w =
x =
y =
z =
```

What gets printed?

```
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?

```
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)

```
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 );
  }
}
```

Output

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?

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

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

 9      ref.method1( Test3.a );
10    }

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

15    private void method1( int x )
16    {
17      int c = x + 4;
18      int b;

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

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( "x = " + x );
28      System.out.println( "result = " + method2( 11 ) );
29      System.out.println( "Test3.a = " + Test3.a );
30      System.out.println( "this.b = " + this.b );
31      System.out.println( "this.c = " + this.c );
32      System.out.println( "x = " + x );
33      System.out.println( "a = " + a );
34      System.out.println( "b = " + b );
35      System.out.println( "c = " + c );
36    }

37    public int method2( int x )
38    {
39      int a = x;
40      int c = b;

41      x = b;

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

49      Test3.a = a + 2;
50      this.c = x + c;

51      return x + 5;
52    }
53  }
```

Use the letters below to identify various program parts.

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

_____ `5` on line 8

_____ `main()` on line 6

_____ `Test3()` on line 11

_____ `Test3` on line 1

_____ `method1()` on line 15

_____ `b` on line 18

_____ `x` on line 37

_____ `c` on line 40

_____ `b` on line 4

_____ `a` on line 3

Output

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

4)

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 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;
}
```

Output

5) Given the following definitions:

```
public interface Printable
{
  public abstract String print( boolean duplex );
}
```

```
class Thing1 implements Printable
{
  private String str;

  public Thing1()
  {
    this.str = "Thing 1";
  }

  public String print( boolean duplex )
  {
    return this.str + " duplex = " + duplex;
  }

  public String print()
  {
    // print single sided by default
    return this.print( false );
  }
}
```

```
class Thing2 implements Printable
{
  private String str;

  public Thing2()
  {
    this.str = "Thing 2";
  }

  public String print( boolean duplex )
  {
    return this.str + " duplex = " + duplex;
  }

  public String print( String user )
  {
    System.out.print( user + ": " );

    // print double sided by default
    return this.print( true );
  }
}
```

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".

```
printable = thing1;

System.out.println( printable.print( true ) );     _____

System.out.println( thing1.print() );              _____

System.out.println( printable.print() );           _____

printable = thing2;

System.out.println( printable.print( "CS11FZZ" ) ); _____

System.out.println( printable.print( false ) );    _____

System.out.println( thing2.print( "CS11FZZ" ) );   _____
```

What two <u>additions</u> 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 <u>added</u> to which file(s). Do not remove or change any of the existing code.

1)

2)

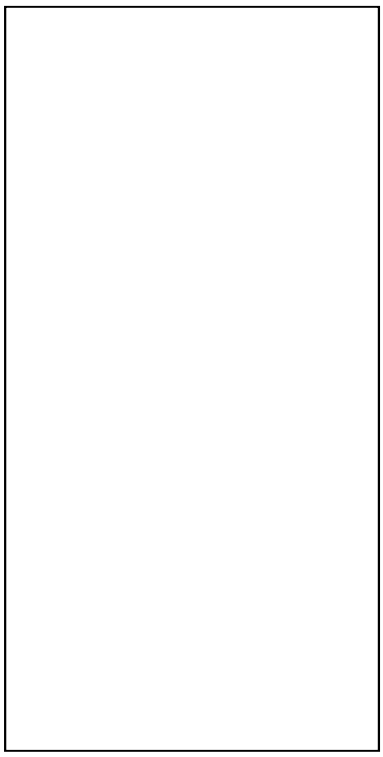
6) Trace the following program and specify its output.

```
public class Trace
{
  public static void main( String[] args )
  {
    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 `!` )

```
!( 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

```
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

`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

**Scratch Paper**