Signature ______________________ **Name** __________________________

cs11f ____ **Student ID** _____________________

By filling in the above and signing my name, I confirm I will complete this exam with the utmost integrity and in accordance with the Policy on Integrity of Scholarship.

CSE 11
Final
Fall 2013

Page 1 ___________ (13 points)

Page 2 ___________ (29 points)

Page 3 ___________ (29 points)

Page 4 ___________ (20 points)

Page 5 ___________ (8 points)

Page 6 ___________ (12 points)

Page 7 ___________ (17 points)

Page 8 ___________ (35 points)

Page 9 ___________ (11 points)

Page 10 ___________ (11 points)

Page 11 ___________ (23 points)

Total ___________ (208 points = 198 base points + 10 points EC)
(100%) [>5%]

This exam is to be taken by yourself with closed books, closed notes, no electronic devices.
You are allowed both sides of an 8.5"x11" sheet of paper handwritten by you.

(Partial) Operator Precedence Table

<table>
<tr><th>Operators</th><th>Associativity</th></tr>
<tr><td>! ++ -- (pre & post inc/dec)</td><td>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>left to right</td></tr>
<tr><td>=</td><td>right to left</td></tr>
</table>

1) What is stored in the memory location allocated for the variable x for the following: _______

```
int x = -99;
```

A) x B) the value -99 C) int

D) a reference (address in memory) to an object which has the value -99 stored

2) What is printed by the following code?

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

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

3) What is stored in the memory location allocated for the variable x for the following: _______

```
String x = "-99";
```

A) x B) the value "-99" C) String

D) a reference (address in memory) to an object which has the characters "-99" stored

4) What are the values of the indicated variables after the following code segments are executed? Remember short-circuit evaluation with && and ||.

```
int a = 7, b = 3, c;
boolean bool1 = !(b > 6) && (a >= 3) && (a <= 4) || (b > 6);

if ( a++ >= 4 && --b >= 2 )
  c = ++a + b--;
else
  c = a++ + --b;
```

a =

b =

c =

bool1 =

```
int x = 7, y = 3, z;
boolean bool2 = !((x > 4) && (y <= 6)) == ((y <= 4) || (x > 6));

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

x =

y =

z =

bool2 =

5) What gets printed?

```
public class Question5
{
  public static void main( String[] args )
  {
    final int MAX = 7, MIN = 2;
    int i = 2, j = 5;

    while ( i < MAX )
    {
      while ( j >= MIN )
      {
        --j;
        System.out.println( i + " " + j );
        j -= 2;
      }
      j = i;
      i++;
    }

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

6) What gets printed by this code?

```
public class Number6
{
  public static void main( String[] args )
  {
    int[] array = { 35, 95, 125, 105, 125, 105, 65, 35, 95, 65 };

    int a = 42, b = 0;

    for ( int i = 0; i < array.length; ++i )
    {
      int foo = array[i];

      if ( foo > a )                    // line that changes in questions below
      {
        a = foo;
        b = i;
      }
    }

    System.out.println(a + ", " + b);  // separate the value of a and b with a comma and space
  }
}
```

Put your answer here: __________

What is printed if the line `if ( foo > a )` was changed to `if ( foo <= a )`? __________

What is printed if the line `if ( foo > a )` was changed to `if ( foo >= a )`? __________

What is printed if the line `if ( foo > a )` was changed to `if ( foo < a )`? __________

What is printed if the line `if ( foo > a )` was changed to `if ( foo == a )`? __________

What is printed if the line `if ( foo > a )` was changed to `if ( foo != a )`? __________

7) In the statement

```
    g.drawOval( 30, 30, 50, 50 );
```

the arguments represent ______
(write the letter representing your answer in the blank above.)

A) Center point of oval and x diameter and y diameter
B) Upper left corner and lower right corner of bounding box
C) Center point of oval and width and height of bounding box
D) Upper left corner and width and height of bounding box
E) Center point of oval and x radius and y radius

8) What is the output of the following program?

```
public class Tricky
{
  public static void main( String[] args )
  {
    System.out.println( "I" );
    message1( "Java" );
    System.out.println( "II" );
    message2( "Finals" );
    System.out.println( "III" );
    message3( "Fall" );
  }

  public static void message1( String s )
  {
    System.out.println( s + "-0" );
  }

  public static void message2( String s )
  {
    System.out.println( s + "-1" );
    message1( s + "-2" );
    System.out.println( s + "-3" );
  }

  public static void message3( String s )
  {
    System.out.println( s + "-4" );
    message2( s + "-5" );
    System.out.println( s + "-6" );
  }
}
```

9) Which part of the method mystery() below is the base case (part labeled A or B)? _____
Which part of the method mystery() below is the recursive case (part labeled A or B)? _____

What is printed when this program is run? Drawing stack frames for each method call will probably help.

```
public class Test9
{
  public static void main( String[] args )
  {
    System.out.println( mystery( 6 ) );  // Print returned value
  }

  public static int mystery( int n )
  {
    int result;

    if ( n > 1 )   // A
    {
      result = 2 * n - 1 + mystery( n - 1 );
      System.out.println( n + ": " + result );
    }
    else           // B
    {
      result = 1;
      System.out.println( n + ": " + result );
    }

    return result;
  }
}
```

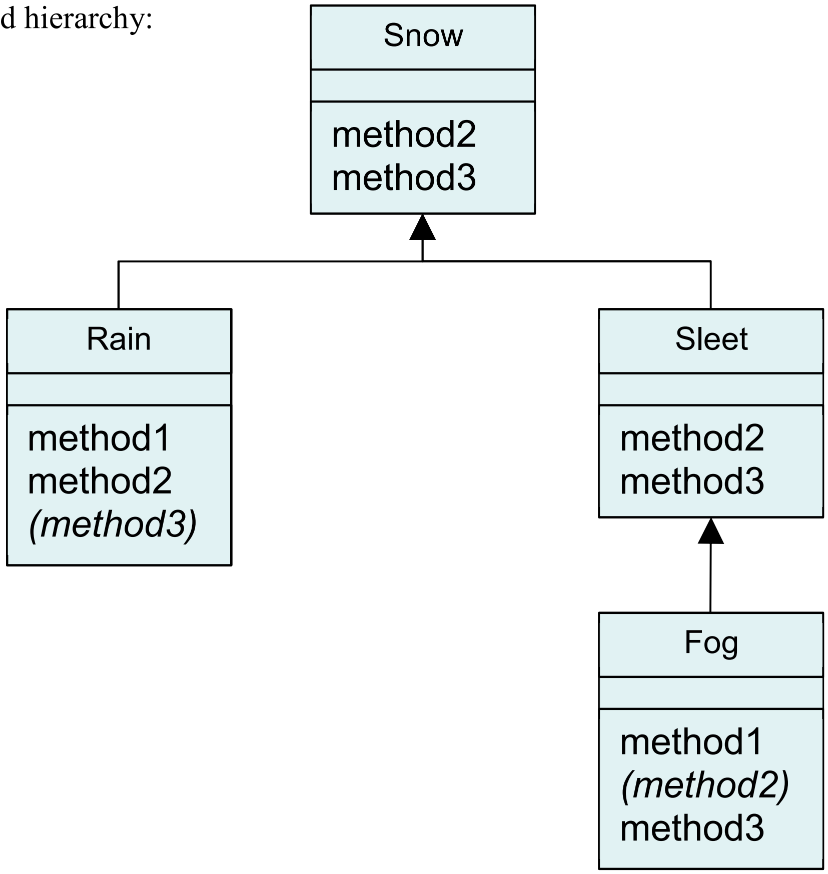
Output

10) What gets printed?

```
int a = 3;
int b = 5;
int c = 7;

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

11) Given the following class definitions and hierarchy:

```
class Snow
{
  public void method2()
  {
    System.out.println("Snow 2");
    method3();
  }

  public void method3()
  {
    System.out.println("Snow 3");
  }
}

class Rain extends Snow
{
  public void method1()
  {
    method3();
    System.out.println("Rain 1");
  }

  public void method2()
  {
    method3();
    System.out.println("Rain 2");
    super.method2();
  }
}

class Sleet extends Snow
{
  public void method2()
  {
    method3();
    System.out.println("Sleet 2");
    super.method2();
  }

  public void method3()
  {
    System.out.println("Sleet 3");
  }
}

class Fog extends Sleet
{
  public void method1()
  {
    System.out.println("Fog 1");
  }

  public void method3()
  {
    System.out.println("Fog 3");
    super.method3();
  }
}
```



What is the output given the following code:

```
Snow ref1;

ref1 = new Fog();

((Fog) ref1).method1();
System.out.println( "-----" );
ref1.method2();
System.out.println( "-----" );
ref1.method3();
```

Put your answer here:

____

____

____

____

____

____

____

____

____

____

____

What is the output given the following code:

```
Snow ref1;

ref1 = new Rain();

((Rain) ref1).method1();
System.out.println( "-----" );
ref1.method2();
System.out.println( "-----" );
ref1.method3();
```

Put your answer here:

____

____

____

____

____

____

____

____

____

12)

What gets printed by the following code? ________

```
int x = 13;
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 = 13;
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 = 13;
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 = 13;
if ( x > 7 )
{
  x += 2;  // Same as x = x + 2;
}
else if ( x >= 10 )
{
  x += 6;
}
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 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 );
```

13) 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:

```
Thing1 thing1 = new Thing1();
Thing2 thing2 = new Thing2();
Printable printable;
```

<u>Hint</u>: What does the compiler know about any reference variable at compile time (vs. run time)?

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" and assume that line is commented out when run.

```
System.out.println( thing1.print() );                ______________________________________________

System.out.println( thing1.print( true ) );          ______________________________________________

System.out.println( thing1.print( "CS11FZZ" ) );     ______________________________________________

System.out.println( thing2.print() );                ______________________________________________

System.out.println( thing2.print( true ) );          ______________________________________________

System.out.println( thing2.print( "CS11FZZ" ) );     ______________________________________________

printable = thing1;

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

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

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

printable = new Thing2();

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

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

System.out.println( printable.print( "CS11FZZ" ) );  ______________________________________________
```

14) Using only the statements below, select the order of the statements to draw an E such that the width of the E is `size` pixels and the height of the E is twice `size` pixels. Do not worry about where it is drawing. Assume the turtle is pointing up when the method is called, the pen is down, and it is positioned at the upper left corner of where we want to draw the E. Start drawing the E at the upper left corner of the E. Have the turtle end at the bottom right corner of the E.

Write the letter corresponding to each statement in the correct order to draw an E. Do it in exactly 12 statements.

```
A) this.forward( size );
B) this.turn( 90 );  // right
C) this.forward( -size );
D) this.turn( -90 ); // left
```

```
public void drawE( int size )
{
  _____

  _____

  _____

  _____

  _____

  _____

  _____

  _____

  _____

  _____

  _____

  _____
}
```

15) What is the equivalent Java expression for the following expression such that no `!` operators are used?

`!( x > 42 && y != 37 )` ____

What gets printed if the value of the actual argument passed to this method is 5? _______

```
public void m6( int x )
{
   int y = 0;

   if ( x <= 1 )
      y = 3;
   else if ( x <= 2 )
      y = 5;
   else if ( x == 3 || x >= 4 )
      y = 7;
   else
      y = 9;

   System.out.println( y );
}
```

What gets printed if the value of the actual argument passed to this method is 2? _______

```
public void m6( int x )
{
   int y = 0;

   if ( x <= 1 )
      y = 3;
   if ( x <= 2 )
      y = 5;
   if ( x == 3 || x >= 4 )
      y = 7;
   else
      y = 9;

   System.out.println( y );
}
```

16) Consider the following program?

```
 1  public class Test16
 2  {
 3    private int a;
 4    private int b;
 5    private static int c = 5;

 6    public static void main( String[] args )
 7    {
 8      Test16 ref = new Test16( 2 );

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

11    public Test16( int a )
12    {
13      this.a = a;
14    }

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

19      b = a;
20      a = c;

21      System.out.println( "this.a = " + this.a );
22      System.out.println( "this.b = " + this.b );
23      System.out.println( "Test16.c = " + Test16.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( a ) );
28      System.out.println( "this.a = " + this.a );
29      System.out.println( "this.b = " + this.b );
30      System.out.println( "Test16.c = " + Test16.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    private int method2( int x )
37    {
38      int b = x;
39      int c = this.b + Test16.c;

40      x = a = b + c;

41      System.out.println( "this.a = " + this.a );
42      System.out.println( "this.b = " + this.b );
43      System.out.println( "Test16.c = " + Test16.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      Test16.c = c + 2;
49      this.a = a + c;

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

Use the numbers below to identify various program parts.

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

_____ `Test16()` on line 11 _____ `c` on line 5

_____ `2` on line 8 _____ `a` on line 11

_____ `main()` on line 6 _____ `x` on line 17

_____ `Test16` on line 1 _____ `a` on line 3

_____ `method1()` on line 15 _____ `c` on line 17

Where in the Java Runtime environment does each of the following live?

`c` on line 39 ___________ `a` on line 3 ____________

`a` on line 11 ___________ `c` on line 5 ____________

Output

```
this.a = ________
this.b = ________
Test16.c = ________
c = ________
b = ________
a = ________
this.a = ________
this.b = ________
Test16.c = ________
x = ________
a = ________
b = ________
c = ________
result = ________
this.a = ________
this.b = ________
Test16.c = ________
x = ________
a = ________
b = ________
c = ________
```

Given the following class definitions for class Foo, class Fubar, and class FubarTest:

```
public class Foo
{
  public Foo()
  {
    this( 42, 420 );
    System.out.println( "Foo ctor #1" );
  }

  public Foo( int x, int y )
  {
    System.out.println( "Foo ctor #2" );
  }

  public String toString()
  {
    System.out.println( "Foo" );
    return "Foo.toString";
  }
}
```

```
public class Fubar extends Foo
{
  public Fubar( int x, int y, int z )
  {
    this();
    System.out.println( "Fubar ctor #1" );
  }

  public Fubar( int x, int y )
  {
    this( x, y, 4200 );
    System.out.println( "Fubar ctor #2" );
  }

  public Fubar()
  {
    System.out.println( "Fubar ctor #3" );
  }

  public String toString()
  {
    System.out.println( "Fubar" );
    String s = "Fubar" + " + " +
               super.toString();
    return s;
  }
}
```

```
public class FubarTest
{
  public static void main( String[] args )
  {
    Foo ref = new Fubar( 42, 420 );

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

    System.out.println( ref.toString() );
  }
}
```

17) What is the output when we run FubarTest as in
java FubarTest

Given the initial order of ints in an array as: 4, 7, 10, 9, 1, 2, 6 what is the order of the elements after 3 iterations of the selection sort algorithm? Recall the selection sort algorithm finds the index of the smallest value in the unsorted partition and exchanges (swaps) that value with the value at the index of the first element of the unsorted partition, then increments the index of the unsorted partition.

____ ____ ____ ____ ____ ____ ____

What Java annotation did we use for methods like equals() and toString() in subclasses to ensure the same signature was being used in the subclass as was defined in the superclass? ______________________________

18) Given the definition of class Swap below, indicate the output of each println statement?
(Hint: Draw stack frames)

```
public class Swap
{
  private int a;

  public int getA()
  {
    return a;
  }

  public void setA(int a)
  {
    this.a = a;
  }

  public Swap(int a)
  {
    this.a = a;
  }

  public int swap(int a)
  {
    this.a = a;

    return a;
  }

  public void swap(int a, int b)
  {
    int tmp;

    tmp = a;
    a = b;
    b = tmp;
  }

  public void swap(Swap ref)
  {
    Swap tmp;

    tmp = ref;
    ref.a = this.a;
    this.a = tmp.a;
  }

  public Swap swap(Swap ref, int a)
  {
    this.a = a;

    return ref;
  }

  public static void swap(Swap ref1, Swap ref2)
  {
    Swap tmp;

    tmp = ref1;
    ref1 = ref2;
    ref2 = tmp;
  }
}
```

```
public class SwapTest
{
  public static void main( String[] args )
  {
    int a = 42;  Swap ref1;
    int b = 64;  Swap ref2;

    ref1 = new Swap(7);
    ref2 = new Swap(2);

    ref2 = ref1.swap(ref2, a);

    System.out.println(ref1.getA());  _________

    System.out.println(ref2.getA());  _________


    ref1 = new Swap(7);
    ref2 = new Swap(2);

    ref1.setA(ref1.swap(ref2.getA()));

    System.out.println(ref1.getA());  _________

    System.out.println(ref2.getA());  _________


    ref1 = new Swap(7);
    ref2 = new Swap(2);

    Swap.swap(ref1, ref2);

    System.out.println(ref1.getA());  _________

    System.out.println(ref2.getA());  _________


    ref1 = new Swap(7);
    ref2 = new Swap(2);

    ref1.swap(a, b);

    System.out.println(a);            _________

    System.out.println(b);            _________


    ref1 = new Swap(7);
    ref2 = new Swap(2);

    ref1.swap(ref2);

    System.out.println(ref1.getA());  _________

    System.out.println(ref2.getA());  _________
  }
}
```

The different swap() method definitions have the same name but differ in their formal parameters. This is an example of method ________________________ .

19) 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? ____________

Assume a program had the following definitions (a Point has an x and a y value):

```
    Point p1 = new Point( 420, 42 );
    Point p2 = new Point( p1 );
    Point p3 = p2;
```

What results would be produced by evaluating the following expressions?

```
p1 == p2       ____________      p1 == p3       ____________      p2 == p3       ____________

p1.equals(p2) ____________       p1.equals(p3) ____________       p2.equals(p3) ____________

p3.translate(1, 1);  // Add 1 to the x and y coordinates in the Point object ref'ed by p3

p1.equals(p2) ____________       p1.equals(p3) ____________       p2.equals(p3) ____________
```

You type `java Foo2` at the command line and you get the following:

```
Exception in thread "main" java.lang.NumberFormatException: For input string: "123b5"
        at java.lang.NumberFormatException.forInputString(NumberFormatException.java:63)
        at java.lang.Integer.parseInt(Integer.java:490)
        at java.lang.Integer.parseInt(Integer.java:531)
        at FooBar.foo2(BarNone.java:69)
        at Foo2.main(Foo2.java:28)
```

Is this a compile time or a run time error? ________________________

What is the value of the string we were trying to convert to an int? _____________________

What method in what class in what file and line number in your code did this occur?

Method _______________________________

Class _______________________________

File _______________________________

Line # _______________________________

Regarding the Snow, Rain, Sleet, Fog program on page 4 (#11),

class Rain is ______ method3() from class Snow

class Rain is ______ method2() from class Snow

A) overriding
B) overwriting
C) overloading
D) inheriting
E) finalizing
G) abstracting
H) all of the above
I) none of the above

Regarding class Snow on page 4 (#11), specify the two things the Java compiler will automatically insert into the resulting Snow.class bytecode file. Be specific. Write code.

1)

2)

Scratch Paper

Scratch Paper