Signature ______________________ **Name** __________________________

cs11f ____ **Student ID** ____________________

CSE 11
Final
Fall 2008

Page 1 ___________ (20 points)

Page 2 ___________ (10 points)

Page 3 ___________ (16 points)

Page 4 ___________ (28 points)

Page 5 ___________ (7 points)

Page 6 ___________ (26 points)

Page 7 ___________ (16 points)

Page 8 ___________ (13 points)

Page 9 ___________ (10 points)

Total ___________ (146 points = 140 base points + 6 points EC [5%])
(100%)

(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 are not valid Java identifiers? (Circle your answer(s).)

```
thirdLine          3rdLine            line3              extends

S.E.               South East         _SE_               Public
```

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

```
int i = 1, j = 2, k = 3, m = 2;

System.out.println( j >= i && k == m );      ________

System.out.println( i >= 1 || j < 4 );       ________

System.out.println( !( k > m ) );            ________

System.out.println( !((i > 4) || (j <= 6)) == ((i <= 4) && (j > 6)) );     _________
```

3) What gets printed?

```
int a = 2, b = 7;

System.out.println(  -1  +  ++a  *  5  +  17  %  5  );       _________

System.out.println(  6  +  b++  -  5  /  9  +  4  );         _________
```

4) What gets printed?

```
public class Question4
{
   public static void main( String[] args )
   {
      final int MAX = 6;
      int i = -2, j = -3;

      for (i = 4; i < MAX; ++i )
      {
         j = 5;
         while  ( j <= MAX )
         {
            System.out.println( i + " " + j );
            ++j;
         }
      }

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

5) What gets printed as a result of the call `F5( 3, -1 )`? __________

```
public void F5( 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 F5() that would result in the method printing "D".

```
      F5( _____ , _____ );
```

6) What value is returned as a result of the call `a(6)` to the method below? ________

```
  public int a( int x )
  {
    int local;

    if ( --x > 0 )
    {
      local = a( x - 1 ) + x;
      return local;
    }
    else
    {
      return 0;
    }
  }
```

Which of the following rewrites of this method are the same as the one above? A B C or All of them
Circle your answer at the end of the line above.

A

```
public int a( int x )
{
  if ( --x > 0 )
    return a( x - 1 ) + x;
  else
    return 0;
}
```

B

```
public int a( int x )
{
  int local = 0;

  if ( --x > 0 )
    local = a( x - 1 ) + x;

  return local;
}
```

C

```
public int a( int x )
{
  if ( --x > 0 )
    return a( x - 1 ) + x;

  return 0;
}
```

7) A bear is an animal and a zoo contains many animals, including bears. Three classes Animal, Bear, and Zoo are declared to represent animal, bear, and zoo objects. Which of the following is the most appropriate set of declarations? ________

A

```
class Animal extends Bear { ... }
class Zoo {
  private Animal[] zooAnimals;
}
```

B

```
class Bear extends Animal { ... }
class Zoo {
  private Animal[] zooAnimals;
}
```

C

```
class Animal extends Zoo {
  private Bear zooBear;
}
```

D

```
class Bear extends Animal, Zoo { ... }
```

E

```
class Zoo extends Animal {
  private Bear[] zooAnimals;
}
```

F

```
class Bear extends Animal implements Zoo { ... }
```

8) An interface definition is limited to having only ______________________________ and

________________________________.

A concrete class cannot have ___________________________________ declared in its definition.

A _____________ class cannot have any subclasses.

The keyword to denote inheritance of interface is ______________________.

The keyword to denote inheritance of implementation is ______________________.

9) Complete the following method which is intended to return the largest integer in the array numbers.

```
public static int findMax( int[] numbers )     // Assume length of numbers > 0
{
  int positionOfMax = 0;

  for ( int index = 1; _____________________________________; ++index )
  {
    if ( ______________________________________________________ )

      _______________________________________________________ ;
  }
  return numbers[positionOfMax];
}
```

10) What does the following method print as a result of the call `F10( 5 )`?

```
public void F10( int x )
{
  if ( x > 1 )
  {
    F10( x - 1 );
  }

  for ( int y = 1; y <= x; ++y )
  {
    System.out.print( y + " " );
  }

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

11) Indicate whether each of the following parts of a Java program is (A-H) and where in the Java Runtime Environment this lives (1-3)

A) Class (static) variable
B) Instance variable
C) Local variable
D) Parameter
E) Constructor
F) Static method
G) Instance method
H) Class definition

1) Class Area
2) Object in the Heap
3) Stack Frame in the Runtime Stack

Java program part
(Answer A-H in this column)

Java Runtime area
(Answer 1-3 in this column)

```
public class F11                                         ________      F11
{
   public static char actor;                             ________      actor                     ________

   private double elected;                               ________      elected                   ________

   public double getElected() { return elected; }        ________      getElected                ________
}


public class SomeOtherClass                              ________      SomeOtherClass
{
   public int digit;                                     ________      digit                     ________

   public static void main( String[] args ) {            ________      main                      ________

                                                         ________      args                      ________

               ...     // *** Location 1 ***
      F11 ref1;                                          ________      ref1                      ________

      ref1 = new F11();                                         (where ref1 is pointing)         ________

      SomeOtherClass ref2 = new SomeOtherClass();        ________      ref2                      ________

               ...     // *** Location 2 ***                    (where ref2 is pointing)         ________
   }

   public boolean foo( char test ) { ... }               ________      foo                       ________

                                                         ________      test                      ________
}
```

Write a statement that could appear above at the line marked `//*** Location 1 ***` that prints the value of **actor** in class F11.

Write a statement that could appear above at the line marked `//*** Location 2 ***` that assigns the value of **elected** in the F11 object referenced by **ref1** into the variable **digit** in the SomeOtherClass object referenced by **ref2**. Note: elected is private. Also note the types of elected and digit.

12) Given the following partial class definition fill in the body of the constructors using the supplied comments as a guide.

```
public class Page2 extends Page1
{
  private int var1;
  private String var2;

  public Page2( int var1, String var2, double var3 )
  {

    ______________________________            // Explicitly invoke super class (Page1) constructor
                                              //   passing the parameter var3.

    ______________________________            // Initialize the int instance variable to the
                                              //   parameter var1.

    ______________________________            // Initialize the String instance variable to the
                                              //   parameter var2.
  }

  public Page2()
  {
                                              // Call same class constructor passing in 3 for var1,
    _______________________________           //   "Hello" for var2, and 4.20 for var3.

  }

}
```

Assuming class Page1 has only one constructor, and based on the comments and your code above, write the full constructor that must be in class Page1.

```
public class Page1
{
  double var3;










}
```

13) Which of the following is a reason to use an ArrayList instead of an array? _______

A) An ArrayList allows faster access to the ith item than an array does.
B) An ArrayList always uses less memory than an array does.
C) An ArrayList can store objects and an array can only store primitive types.
D) An ArrayList resizes itself as necessary when items are added, but an array does not.
E) An ArrayList provides access to the number of items it stores, but an array does not.

14) The following method is intended to return true if x is between lower1 and upper1, inclusive, or between lower2 and upper2, exclusive, and false otherwise. You can assume lower1 <= upper1 and lower2 <= upper2. Write the body of the method below. Do not define any local variables. You can do this in a single statement.

```
public boolean between( int x, int lower1, int upper1, int lower2, int upper2 )
{




}
```

15) Complete the following method to satisfy the following (you can assume the length of array is > 0):

The value of n1 when output is the sum of all positive even values in array.

The value of n2 when output is the sum of all negative values in array.

```
public void method( int[] array )
{
  int n1 = 0;
  int n2 = 0;





  System.out.println( n1 );
  System.out.println( n2 );
}
```

16) Using methods `getAA()` and `getBB()` and accessing B17's String representation, complete the one line body of the `toString()` below.

```
public class F16 extends B17
{
  private int aa;
  private Location bb;

  /*
   * Returns the String consisting of bb's String representation and B17's String
   * representation and aa each separated by a space (in that order).
   */
  public String toString()
  {
    ______________________________________________________________________________ ;
  }

  /* Other methods including getAA() and getBB() */
}
```

17) In HW14, class Circle was defined to have a center Point and an int radius along with a few constructors and various methods. Complete the following copy constructor and simplified draw() method for class Circle making sure you do not directly access the center and radius members – use the appropriate accessor/mutator methods defined with the class as part of the assignment.

```
public abstract class Shape
{
  private String name;

  public Shape( String name )
  {
    this.setName( name );
  }

  public abstract void draw( DrawingCanvas canvas );

  // Other constructors and methods of class Shape assumed here.

} // end class Shape


public class Circle extends Shape
{
  private Point center;
  private int radius;

  /*
   * Define copy constructor here (use Point's copy constructor, not simple assignment).
   */
  public Circle( Circle c )
  {









  } // end Circle copy ctor

  /*
   * Define draw method here.
   * Assume color set elsewhere. Just draw the filled shape with the appropriate
   * objectdraw library method with 5 args: x, y, width, height, canvas.
   *
   * Note: The circle should be centered around the center Point! Do not use a move() method.
   */
  public void draw( DrawingCanvas canvas )
  {











  } // end draw()

  // Other constructors and methods of class Circle assumed here, including accessors/mutators.

} // end class Circle
```

```
abstract class Animal
{
  private String name;
  public Animal() { this( "Animal" ); }
  public Animal( String name ) { this.name = name; }
  public String toString() { return name; }
  public abstract String speak();
}

class Cat extends Animal
{
  public Cat() {}
  public Cat( String name ) { super( name ); }
  public String speak() { return "Meow"; }
}

class Tiger extends Cat
{
  public Tiger() { this( "Tigger" ); };
  public Tiger( String name ) { super( name + " Tiger" ); }
  public String speak( String name ) { return name + " Roar"; }
}

class LittleTiger extends Tiger
{
  public LittleTiger() { super( "Little Tiger" ); }
  public LittleTiger( String name ) { super( name ); }
  public String speak() { return "Little " + super.speak(); }
}

class Lion extends Cat
{
  public String speak() { return "Lion " + super.speak(); }
  public String louder() { return "Louder Lion " + super.speak(); }
}

public class F18
{
  public static void main( String[] args )
  {

    Animal a;

    a = new Cat( "Top Cat" );
    System.out.println( a + " says " + a.speak() );

    a = new Tiger();
    System.out.println( a + " says " + a.speak() );

    a = new LittleTiger( "Tony" );
    System.out.println( a + " says " + a.speak() );

    a = new Lion();
    System.out.println( a + " says " + a.speak() );

  }
}
```

18a) What gets printed when this program is run?

The following will not compile. Rewrite the println() statement so it will compile and run correctly calling the method `louder()` via the reference **a**. You must use the reference **a** to invoke the method `louder()`.

```
    a = new Lion();
    System.out.println( a + " says " + a.louder() );
```

18b) Use the class definitions on the previous page to answer the following:

Can we subclass/extend from Lion like this? Explain why or why not.

```
class LittleLion extends Lion
{
  public LittleLion() { super( "Little Lion" ); }
  public String speak() { return "Little " + super.speak(); }
}
```

Can we subclass/extend from Animal like this? Explain why or why not.

```
class Dog extends Animal
{
  public Dog() { super( "Dog" ); }
  public String speak( String name ) { return name + " says Woof"; }
}
```

If class Cat was defined as a final class (`final class Cat extends Animal`), can we define SuessCat like this? Explain why or why not.

```
class SuessCat extends Cat
{
  public String toString() { return "Cat in the Hat " + super.toString(); }
}
```

Can we make `abstract class Animal` an interface instead of a class (`interface Animal`) and change `class Cat extends Animal` to `class Cat implements Animal`? Explain why or why not.

If you were writing this exam, what two serious questions would you put on this exam?

1)

2)

Scratch Paper

Scratch Paper