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

<table>
<thead>
<tr>
<th>Operators</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<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></td>
</tr>
<tr>
<td>=</td>
<td>right to left</td>
</tr>
</tbody>
</table>

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 `||`.

```java
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?

```java
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?

```java
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 \texttt{F5( 3, -1 )}? \underline{\hspace{2cm}}

```java
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".

\texttt{F5( \_\_\_\_ , \_\_\_\_ );}

6) What value is returned as a result of the call \texttt{a(6)} to the method below? \underline{\hspace{2cm}}

```java
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.

\begin{tabular}{|c|}
\hline
A \hspace{2cm} B \hspace{2cm} C \\
\hline
public int a( int x )
{
    if ( --x > 0 )
    {
        return a( x - 1 ) + x;
    }
    return 0;
}
\hline
public int a( int x )
{
    int local = 0;
    if ( --x > 0 )
    {
        local = a( x - 1 ) + x;
        return local;
    }
    return 0;
}
\hline
public int a( int x )
{
    if ( --x > 0 )
    {
        return a( x - 1 ) + x;
    }
    return 0;
}
\hline
\end{tabular}
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  
```java
class Animal extends Bear { ... }  
class Zoo {  
    private Animal[] zooAnimals; 
}
```

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

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

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

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

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.

```java
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 )?

```java
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)

<table>
<thead>
<tr>
<th>Java program part</th>
<th>Java Runtime area</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Answer A-H in this column)</td>
<td>(Answer 1-3 in this column)</td>
</tr>
<tr>
<td><strong>A)</strong> Class (static) variable</td>
<td><strong>I)</strong> Class Area</td>
</tr>
<tr>
<td><strong>B)</strong> Instance variable</td>
<td><strong>I)</strong> Object in the Heap</td>
</tr>
<tr>
<td><strong>C)</strong> Local variable</td>
<td><strong>I)</strong> Stack Frame in the Runtime Stack</td>
</tr>
<tr>
<td><strong>D)</strong> Parameter</td>
<td></td>
</tr>
<tr>
<td><strong>E)</strong> Constructor</td>
<td></td>
</tr>
<tr>
<td><strong>F)</strong> Static method</td>
<td></td>
</tr>
<tr>
<td><strong>G)</strong> Instance method</td>
<td></td>
</tr>
<tr>
<td><strong>H)</strong> Class definition</td>
<td></td>
</tr>
</tbody>
</table>

```java
public class F11 {
    public static char actor;
    private double elected;
    public double getElected() { return elected; }
}

class SomeOtherClass {
    public int digit;
    public static void main( String[] args ) {
        // *** Location 1 ***
        F11 ref1;  
        ref1 = new F11();  
        SomeOtherClass ref2 = new SomeOtherClass();
        // *** Location 2 ***
    }
    public boolean foo( char 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.

```java
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.

```java
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.

```java
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.

```java
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.

```java
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.

```java
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.

```java
class LittleLion extends Lion
{
    public LittleLion() { super( "Little Lion" ); }  // improper super constructor call
    public String speak() { return "Little " + super.speak(); }  // improper super method call
}
```

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

```java
class Dog extends Animal
{
    public Dog() { super( "Dog" ); }  // improper super constructor call
    public String speak(String name) { return name + " says Woof"; }  // method overloads the superclass method
}
```

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.

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

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)