

[P1]

Question:

(a) The superclass constructor always gets executed before the subclass constructor.

Answer:

True

Question:

b - Suppose the Foobar class has been defined with a zero-argument constructor. The statement:

```
System.out.println (new Foobar());
```

will cause an error at execution time unless a toString() method has been defined inside the Foobar class.

Answer:

False - Java automatically applies toString() when using System.out.println

c - You cannot have more than one catch clause per try statement.

Answer:

False - Multiple catch phrases are allowed

Question:

d - A constructor can use the keyword super, as if it were a method name, to invoke a different constructor in the same class.

Answer:

False - Constructor can use super to invoke constructor from superclass

Question:

e - If two String objects are compared using the operator == a runtime error message will occur and the program will abort.

Answer:

False - No runtime error or program abortion but will compare references not content.

```
use if (str1.equals(str2)) {  
    System.out.println("Strings are equal")  
} to compare content.
```

Question:

f - In the base case, a recursive method calls itself with a smaller version of the original problem.

Answer:

```
True - public int factorial(int n) {  
    // Base case  
    if (n == 0) {  
        return 1;  
    } else {  
        // Recursive case
```

```
        return n * factorial(n - 1);  
    }  
}
```

In this example, the base case is when  $n$  is equal to 0, and in this case, the method returns 1. In all other cases, the method makes a recursive call with a smaller problem ( $n - 1$ ), bringing it closer to the base case.

Question:

g - If variable  $a$  is an array, variable  $b$  is an ArrayList, and variable  $c$  is a String object, then the statement

```
System.out.println ( a.length() + " " +  
b.size() + " " + c.length );
```

should compile and execute without producing an error message.

Answer:

True - Arrays do not have length() method. Use a.length instead.