

1. Consider the three classes, List, Queue, and Stack. A Queue is a List, a Stack is a List. Write the UML interconnection diagram to show this class hierarchy.
2. Write the Java code for a class header for the class Stack to show that Stack is a subclass of List. (That is, write the code that appears before the first “{” in the class Stack).
3. Write statements to accomplish all of the following using List, Queue, and Stack as defined in problem 1: declare a variable `list` to be of type List; instantiate an object of type Stack (using no arguments) and make `list` reference that Stack object; instantiate an object of type Queue (using no arguments) and make `list` reference that Queue object; test that `list` references a Stack object and print the words “is a stack” if `list` does reference a Stack object.
4. What is the output of the following code segment?

```
int [] x = new int [5];
x[0] = 4;
for (int i = 1; i < 5; i++)
    x[i] = i * x[i - 1];

for (int j = 0; j < 5; j++)
    System.out.println(x[j]);
```

5. Consider the following code. It will compile.

```
class A {
    private int x;
    public A(int t) {
        x = t;
    }
    public void setX(int x) {
        this.x = x;
    }
    public int getX() {
        return x;
    }
    public String toString() {
        String retStr = "" + x;
        return retStr;
    }
}

class B {
```

```

    public static void main(String[] args)    {
        int x = 5;
        A c = new A(x);
        A b = new A(0);
        b = c;
        b.setX(7);
        /*1*/ System.out.println(x);
        /*2*/ System.out.println(c.getX());
        /*3*/ System.out.println(b);

    }
}

```

When the main method of class B is run,

- (a) What is the output of statement /\*1\*/
  - (b) What is the output of statement /\*2\*/
  - (c) What is the output of statement /\*3\*/
6. We have discussed method overloading and method overriding. Define each of these terms and show an example of each.

7. Write a static method which has an int array as its only formal parameter and which returns the largest value in the array.
8. Write a method called swapTest which has 3 parameters. When invoked by code such as the following:

```
int [] array;  
int pos1, pos2;
```

```
. . .
```

```
swapTest(array, pos1, pos2);
```

swapTest will check that pos1 and pos2 are valid indexes of array, and if so will exchange the elements of array at pos1 and pos2 if the element at pos1 is smaller than the element at pos2.