1. (2 pts) The primary job of a constructor is to initialize the instance variables in a class definition. True/False False If a method in a class definition has the same name as the class, then it is a constructor.

2. (2 pts) Java supports encapsulation, which is the binding together of data and the code which operates on the data, through the class definition, and supports information hiding through access control keywords like public.

3. (2 pts) Explain what it means for an object to be immutable.

If the object is created from a class definition which contains private variables but no public mutator methods, then the object is immutable.

4. (2 pts) What is printed by the following code? Explain the output.

```java
public class Question4 {
    public static void main(String[] args) {
        String temp1 = new String("abc");
        String temp2 = new String("abc");
        String temp3 = "abc";
        String temp4 = "abc";
        System.out.println(temp1 == temp2);
        System.out.println(temp1 == temp3);
        System.out.println(temp1 == temp4);
        System.out.println(temp2 == temp3);
        System.out.println(temp2 == temp4);
        System.out.println(temp3 == temp4);
    }
}
```

false
false
false
false
false
true

When the == operator is used between two reference variables, it determines if those references point to the same object in memory. temp1 and temp2 are created with the new operator which creates two Strings object. temp3 and temp4 are references to String literals which are interned. So only temp3 and temp4 refer to the same object.

5. (2 pts) What does the following print? Explain the output. You may use a calculator or abacus if you want.

```java
public class Question5 {
    public static void main(String[] args) {
        String temp = "This is a test";
    }
}
```
temp.concat("World" + temp.length() + temp.charAt(4) + temp.charAt(5));
temp.concat("T" + temp.charAt(temp.length()/2));
temp.concat("WOW this is alot" + temp.charAt(5));
temp.concat("J" + temp.charAt(temp.length()-4/5-3) + temp.indexOf('s'));
temp.concat("K" + temp.charAt(temp.length()/(int)Math.sin(175*3.14/180)+5));
temp.concat("R" + temp.length()*(int)Math.pow(4,15)*(int)Math.sin(36*temp.length()));
temp.toUpperCase().toLowerCase();
System.out.println(temp.length());
System.out.println(temp);

14
This is a test
Since Strings are immutable, no matter what we do to an instance, we can’t change it.