1. Describe the relationship between an object and its defining class. How do you declare a class? How do you declare an object reference variable? How do you create an object? How do you declare and create an object in one statement?

2. What is wrong with the following programs?
   
   a.  
   ```java
   1 public class ShowErrors {
   2     public static void main(String[] args) {
   3         ShowErrors t = new ShowErrors(5);
   4     }
   5 }
   ```

   b.  
   ```java
   1 public class ShowErrors {
   2     public static void main(String[] args) {
   3         ShowErrors t = new ShowErrors();
   4         t.x();
   5     }
   6 }
   ```

   c.  
   ```java
   1 public class ShowErrors {
   2     public void method1() {
   3         Circle c;
   4         System.out.println("What is radius ");
   5         c = new Circle();
   6     }
   7 }
   ```

   d.  
   ```java
   1 public class ShowErrors {
   2     public static void main(String[] args) {
   3         C c = new C(5.0);
   4         System.out.println(c.value);
   5     }
   6 }
   7 }
   8 class C {
   9     int value = 2;
   10 }
   ```
3. What is the printout of the following code?

```java
public class Foo {
    private boolean x;

    public static void main(String[] args) {
        Foo foo = new Foo();
        System.out.println(foo.x);
    }
}
```

4. Suppose that the class Foo is defined in a. Let f be an instance of Foo. Which of the statements in b are correct?

a.

```java
public class Foo {
    int i;
    static String s;

    void imethod() {
    }

    static void smethod() }
}
```

b. If f is an instance of Foo, which statements are correct?
   i. System.out.println(f.i);
   ii. System.out.println(f.s);
   iii. f.imethod();
   iv. f.smethod();
   v. System.out.println(Foo.i);
   vi. System.out.println(Foo.s);
   vii. Foo.imethod();
   viii. Foo.smethod();

5. What is an accessor method? What is a mutator method? What are the naming conventions for accessor methods and mutator methods?
6. Describe the difference between passing a parameter of a primitive type and passing a parameter of a reference type. Show the output of the following programs:

```java
public class Test {
    public static void main(String[] args) {
        Count myCount = new Count();
        int times = 0;

        for (int i = 0; i < 100; i++)
            increment(myCount, times);

        System.out.println("count is " + myCount.count);
        System.out.println("times is " + times);
    }

    public static void increment(Count c, int times) {
        c.count++;
        times++;
    }
}

public class Count {
    public int count;

    Count(int c) {
        count = c;
    }

    Count() {
        count = 1;
    }
}
```