Tricky Programs

game....

```
// filename Main.java
class Test {
    protected int x, y;
}

class Main {
    public static void main(String args[]) {
        Test t = new Test();
        System.out.println(t.x + " " + t.y);
    }
}
```

Output

0 0

In Java, a protected member is accessible in all classes of same package and in inherited classes of other packages. Since Test and Main are in same package, no access related problem in the above program. Also, the default constructors initialize integral variables as 0 in Java (See this GFact for more details). That is why we get output as 0 0.

```
// filename Test.java
class Test {
    public static void main(String[] args) {
        for(int i = 0; 1; i++) {
            System.out.println("Hello");
            break;
        }
    }
}
```

OUTPUT

- Compiler Error
- There is an error in condition check expression of for loop. Java differs from C++(or C) here.

  C++ considers all non-zero values as true and 0 as false. Unlike C++, an integer value expression cannot be placed where a boolean is expected in Java.

```
// filename Test.java
class Test {
    public static void main(String[] args) {
        for(int i = 0; true; i++) {
            System.out.println("Hello");
            break;
        }
    }
}
```

OUTPUT

Hello

```
class Main {
    public static void main(String args[]) {
        System.out.println(fun());
    }
    int fun() {
        return 20;
    }
}
```

Output

Compiler Error

Like C++, in Java, non-static methods cannot be called in a static method. If we make fun() static, then the program compiles fine without any compiler error.

```
class Main {
    public static void main(String args[]) {
        System.out.println(fun());
    }
    static int fun() {
        return 20;
    }
}
```

OUTPUT

20

```
class Test {
    public static void main(String args[]) {
        System.out.println(fun());
    }
    static int fun() {
        static int x= 0;
        return ++x;
    }
}
```

OUTPUT

- Compiler Error
- Unlike C++, static local variables are not allowed in Java.We can have class static members to count number of function calls and other purposes that C++ local static variables serve.

```
class Test {
   private static int x;
   public static void main(String args[]) {
       System.out.println(fun());
   }
   static int fun() {
       return ++x;
   }
}
```

OUTPUT

- 1