

Testing

A byte size lesson in Java programming.

Why do developers test?

- To find technical mistakes or errors.
- Errors causes the computer or program to not function properly.

Syntax errors

```
String syntaxError = This is a syntax error because " " were not used;  
  
boolean flag = false // forgetting the semi-colon ; is another syntax error  
  
myVariable = 0; // myVariable is not declared and is not a keyword in Java
```

- typos which are usually flagged by the IDE
 - using undeclared variables
 - forgetting colons
 - incorrect statement structure

Which line has the syntax error?

```
1. int i = 1;  
2.  
3. while(i <<< "4") {  
4.     System.out.println(i);  
5.     i++;  
6. }
```

Which line has the syntax error?

```
1. int i = 1;  
2.  
3. while(i < 4) {  
4.     System.out.println(i);  
5.     i++;  
6. }
```

Logical errors

- Trickier, because the code will run but the output will be incorrect
- It usually indicates a mistake in reasoning

```
1. int i = 1;  
2. // the below will never run  
3. while(i > 4) {  
4.     System.out.println(i);  
5.     i++;  
6. }
```


Can you spot the logical error?

```
1. int varA = 5;  
2. int varB = 2;  
3. int mult = varA * varB;  
4. System.out.println("5 added to 2 is" + mult );
```



Can you spot the logical error?

```
1. int area = 0;
2. int breadth;
3. int length;
4.
5. Scanner sc = new Scanner(System.in);
6. System.out.println("Enter the length and breadth");
7. length = sc.nextInt();
8. breadth = sc.nextInt();
9.
10. System.out.println("The area is: " + area);
11. area = length * breadth;
```



Run-time errors

- This is an error that happens when the program is running.
- They are usually because the developer has not considered a situation to occur

```
public static double div(double num1, double num2) {  
    double num3 = num1 / num2;  
    return num3;  
}
```

Most Common Run-time Errors

- Dividing a number by zero
- Accessing an element in an array that is out of range
- Passing an invalid argument to a method
- Attempting to convert an invalid string to a number

Can you spot the run-time error?

```
1. int[] marks = new int[5];  
2. marks[7] = 95;  
3. System.out.println("Mark saved");
```

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: ...

Can you spot the possible run-time error?

```
1. Scanner sc = new Scanner(System.in);  
2. String mark = sc.nextLine();  
3. int[] marks = new int[5];  
4. marks[0] = mark;
```

