Java Assertion

1. Syntax of assert statement
    a. **assert** *expression1*;
    b. **assert** *expression1 : expression2*;
        expression1 must be a **boolean** expression.
        expression2 must return a value (must not return void).
2. How does assertion work?
    a. If assertion is **enabled**, then the assert statement will be evaluated. Otherwise, it does not get executed.
    b. If *expression1* is evaluated to **false**, an **AssertionError** is thrown which causes the program **stops immediately**. And depending on existence of *expression2*:
        i. If *expression2* does not exist, then the
            AssertionError is thrown with no detail error message.
        ii. If *expression2* does exist, then a String representation of
            expression2's return value is used as detail error message.
    c. If *expression1* is evaluated to **true**,
        then the program continues normally.
3. How to enable assertion?

    java -enableassertions Example **OR** java -ea Example

4. Example

```
/**
 * This program, if run with assertion disabled -> OK
 * If run with assertion enabled
 * -> if the number of arguments equals to 5 -> OK
 * -> if the number of arguments is not equal to 5
 *    -> java.lang.AssertionError:
 *          The number of arguments must be 5.
 */
public static void main(String[] args) {
      assert args.length == 5 :
            "The number of arguments must be 5.";
      System.out.println("OK");
}
```

5. Generally, assertion is **enabled** during **development** time to defect and fix bugs, and is **disabled** at **deployment** or **production** to **increase performance**.