Assertion

Ensures the correctness of any assumptions which have been done in the program

Assertions (1.4V)

- Introduction
- assert as keyword and identifier
- Types of assert statements
- Various possible runtime flags
- Assertions Error

Software bug?

A software bug is an error, failure, or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

The debugging process

- In **software development**, debugging involves locating and correcting **code** errors in a computer program. Debugging is part of the **software testing process** and is an integral part of the entire software development lifecycle. The debugging process starts as soon as code is written and continues in successive stages as code is combined with other units of programming to form a software product. In a large program that has thousands and thousands of lines of code, the debugging process can be made easier by using strategies such as **unit tests, code reviews** and **pair programming**.

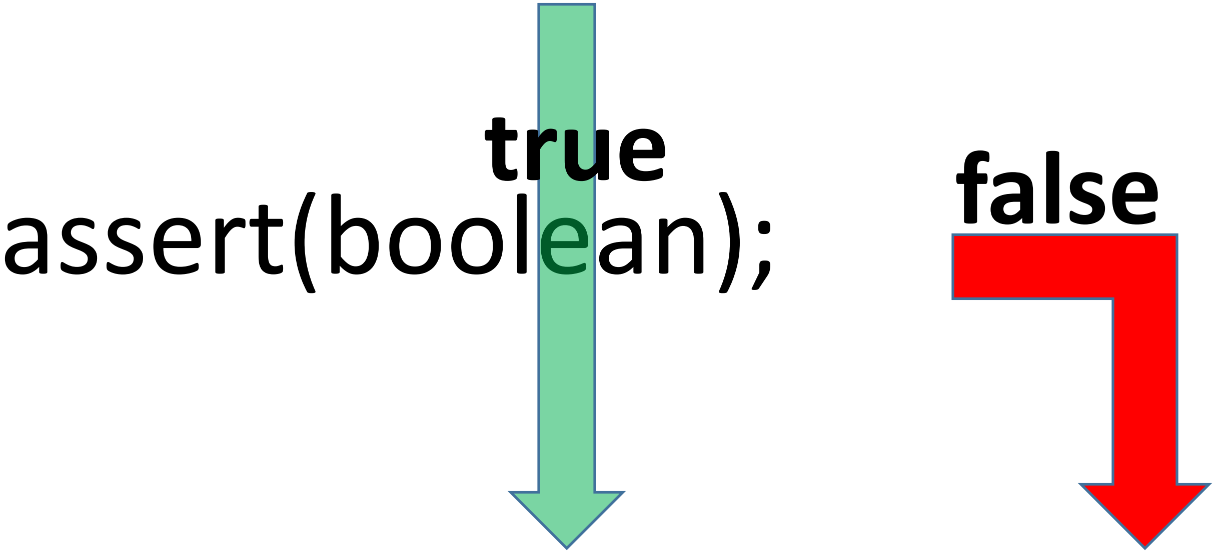
Assertion:

Assertion is a statement in java. It can be used to test your assumptions about the program. It provides an effective way to detect and correct programming errors.

assert $\underbrace{\text{expression;}}_{\text{boolean}}$

```
class A {

    public static void main(String[] args) {
        int i = 10;
        assert i == 11;
        System.out.println("Done");
    }
}
```



true

assert(boolean);

false

- If you use assertion, It will not run simply because assertion is disabled by default. To enable the assertion,

  -ea or **-enableassertions** switch of java must be used.

Types of assert statement

- Simple version
- Augmented version

```
class A {

    public static void main(String[] args) {
        int i = 10;
        assert (i > 10);
        System.out.println("Done");
    }
}
```

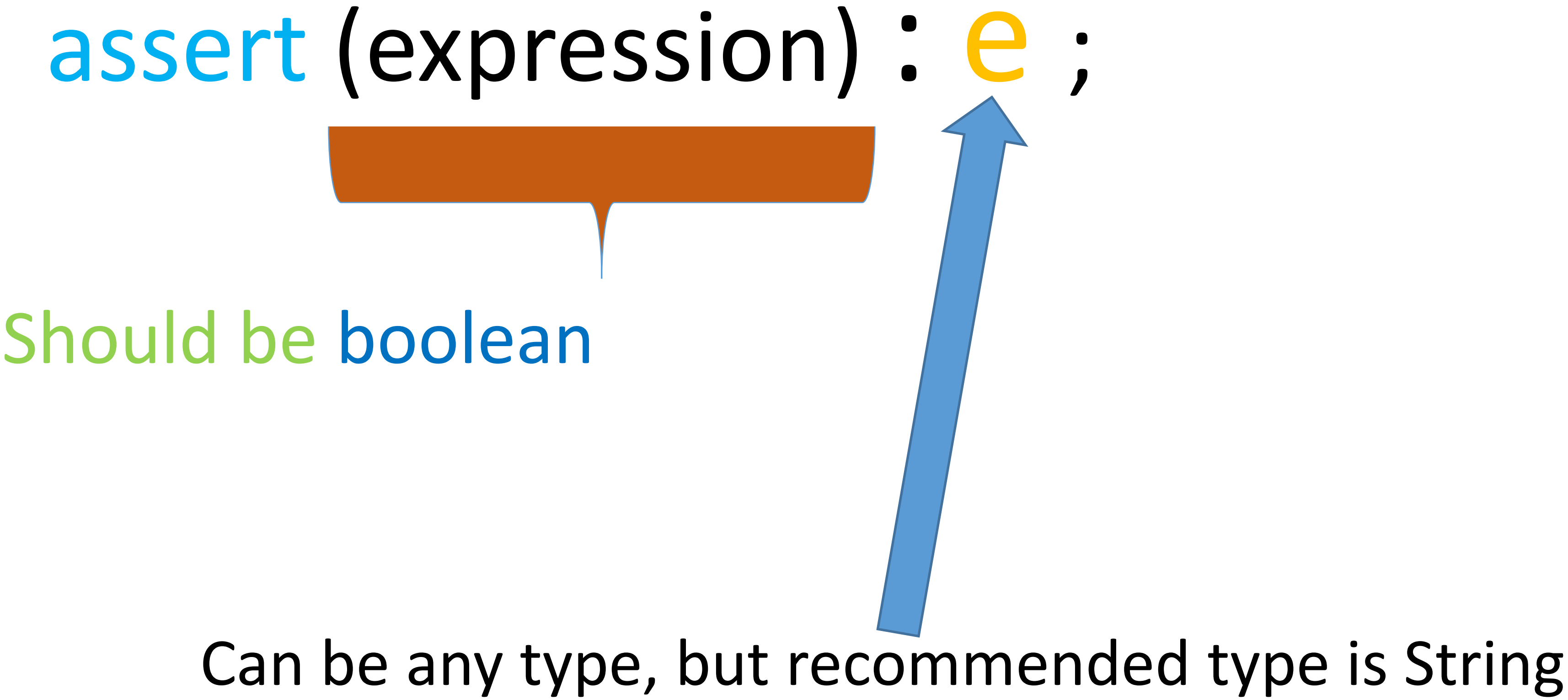
Simple version

```
Exception in thread "main" java.lang.AssertionError
        at A.main(A.java:5)
```

```
class A {

    public static void main(String[] args) {
        int i = 10;
        assert (i > 10) : "Here i value should be > 10, but it is not";
        System.out.println("Done");
    }
}
```

Augmented version



assert (expression) : e ;

Should be boolean

Can be any type, but recommended type is String

```
java  A
```

Vs.

```
java  -ea  A
```

Various possible runtime flags

1. -ea | -enableassertions

   Enable assertions for non-system classes

2. -da | -disableassertions

   Disable assertions for non-system classes

3. -esa | -enablesystemassertions

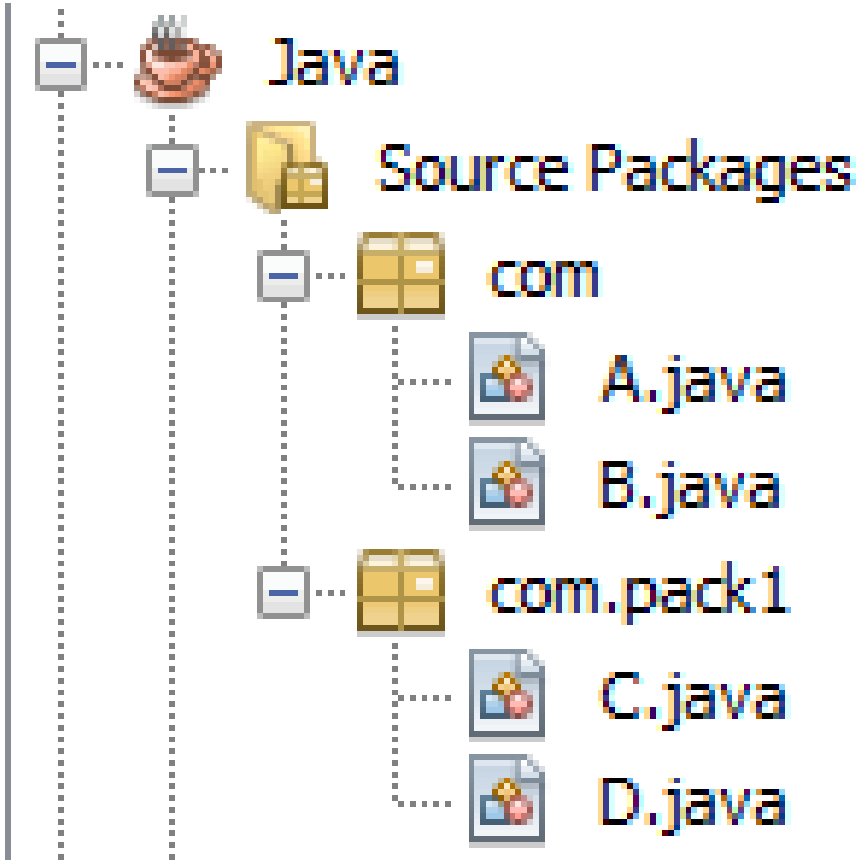
   Enable assertions for system classes

4. -dsa | -disablesystemassertions

   Disable assertions for system classes

```
java -ea -esa -ea -dsa -da -esa -ea -dsa A
```

Simultaneously



```
Java
  Source Packages
    com
      A.java
      B.java
    com.pack1
      C.java
      D.java
```

1. To enable assertions in **B** class ? java -ea:com.B
2. To enable assertions in both **B** and **D** classes ? java -ea:com.B -ea:com.pack1.D
3. To enable assertions in every classes of **com** ? java -ea:com…
4. To enable assertions in every classes of **com** except **B** class? java -ea:com… -da:com.B
5. To enable assertions in every classes of **com** except **pack1** classes ? java -ea:com… -da:com.pack1…