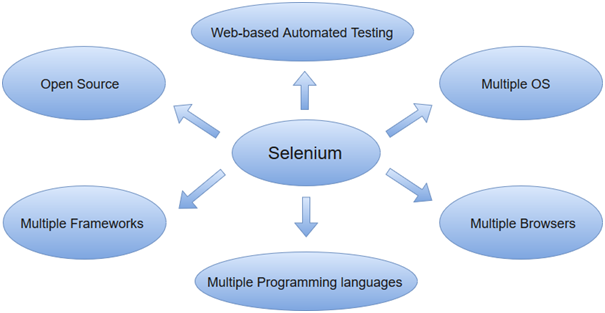
**What is Automation Testing**

Automation testing or test automation uses open source or paid automated testing tools to test [software applications](https://www.hurix.com/how-to-perform-api-testing-on-software-apps/) quickly and efficiently, and 24/7 without human intervention. Automated software testing can reveal the memory contents, data tables, and file contents; this helps to determine if the program is functioning as required. With the advantages of automation framework, one can execute thousands of varied and lengthy test cases that are impossible in manual testing. It can increase the depth of tests to improve software quality.

It is advantageous as it performs the tasks of many manual testers, and improves overall accuracy with little or no manual intervention. As pre-written scripts are tested, this ensures that software of standard quality is released.

Automation testing follows a testing framework approach. The most common ones are keyword-driven framework, data-driven framework, linear scripting framework, and modular testing framework.

Using Test Automation software is the most effective, efficient, and comprehensive approach to enhance your software testing.

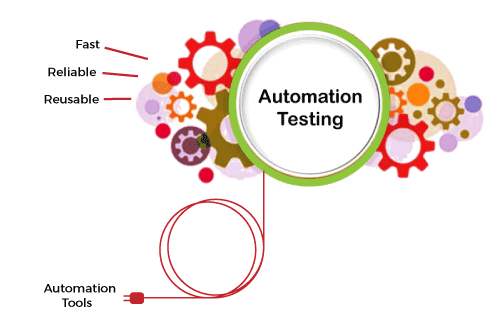


## What Types of Tests Can Be Automated?

Several types of tests can be automated, including:

1. **Unit tests:** These tests focus on individual units of code, such as functions or methods, to ensure that they are working correctly.
2. **Integration tests:** These tests focus on the interactions between different units of code, to ensure that they are working together correctly.
3. **Functional tests:** These tests focus on the overall functionality of the system, to ensure that it is working as expected from the user’s perspective.
4. **Performance tests:** These tests focus on measuring the performance of the system, such as response time and throughput, to ensure that it can handle the expected load.
5. **Acceptance tests:** These tests focus on verifying that the system meets the acceptance criteria defined by the customer or stakeholders.
6. **Security tests:** These tests focus on identifying and addressing security vulnerabilities in the system.
7. **Regression tests:** These tests focus on identifying and addressing regressions, or unintended changes, that occur as a result of code changes.
8. **Smoke tests:** These tests are used to quickly verify that the system is in a stable state after a code change, and is worth proceeding with more detailed testing.

## Here are some of the advantages of automation testing:



The advantages of automation testing are given as follows -

### Saves time

Automating the testing process helps the testing team to use less time to validate newly created features. For instance, in manual testing, there is a need to write thousand test cases for a calculator application, but automation makes the process much faster.

### Productivity improvement

As during execution, automation tests do not require human intervention, so testing an application can be done late at night, and we can get the results next morning. Software developers and testers require less time on automation testing.

### Accuracy improvement

In manual testing, there is a chance of mistakes whether you are an experienced testing engineer. The chances of errors may increase when testing a complex use case. But Automation testing reduces the chances of errors. There is good accuracy, as we will get the same result each time on performing the same test cases.

### Test suite reusability

We can reuse the test scripts in automation testing, and we don't need to write the new test scripts again and again. These test cases can be used in various ways, as they are reusable. Reusability helps to reduce the cost and also eliminate the chances of human error.

### Ability to test on various platforms

Automation testing allows the user to test the application on different web browsers and operating systems.

### Running tests 24/7

In automation testing, we can start the testing process from anywhere in the world and anytime we want. It can also be done remotely if we don't have many approaches or the option to purchase them.

### Early bug detection

By automation testing, it is easy to detect critical bugs in the initial phases of software development. It reduces the cost and helps us to spend fewer working hours to fix such problems. It increases the efficiency of the team.

### Less human resources

Automation testing requires fewer people to perform a tedious manual test. To implement the automation test script, we need a test automation engineer who can write the test scripts to automate our tests.

### Reduce the expenses

Automation testing is less expensive, as once the test scripts have been built, we can reuse them at any time without any extra cost. While manual testing is more expensive than automation, with manual monitoring, it is typical to execute experiments repeatedly.

### Scalability of test cases

In manual testing, we require the involvement of the number of people and number of hours to scale up a project. Whereas the scalability of automation testing is higher, we need adding of test executors to the testing framework.

### Consistency

Compared to manual testing, automation testing is more consistent and way faster than executing the regular monotonous tests that cannot be missed but may cause faults when tested manually.

### Fast development and delivery

Automated tests can be executed repeatedly and completed rapidly. We do not have to wait for weeks to execute the tests; few hours are enough for execution. Switching from manual to automation reduces the waiting time and boosts development.

### Easily execution of lengthy and complicated test cases

Execution of bug-prone and complex test cases is easier with automation testing. Test cases with reproducible steps lead to distraction and wrong assurances on testing them manually.

Some of the other benefits of automation testing are listed as follows -

* In comparison to manual testing, automation testing requires fewer resources.
* It makes load and performance testing, stress testing, and reliability testing possible.
* It is more reliable, as it reduces the occurrence of errors. It is reliable because it tests the application with the help of tools and test scripts.
* With automation testing, test engineers are free to focus on other work.
* It improves the testing coverage as the automatic execution of test cases is faster than manual execution.
* Automation testing allows the execution of test cases in a 24x7 environment.
* It enhances the knowledge of test engineers by producing a repository of different test cases.
* Batch execution is possible using automation testing because all the written scripts can be executed simultaneously.
* Automation testing is 70% faster than manual testing.

What is Java?

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than **3 billion** devices run Java.

It is used for:

* Mobile applications (specially Android apps)
* Desktop applications
* Web applications
* Web servers and application servers
* Games
* Database connection
* And much, much more!

Why Use Java?

* Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
* It is one of the most popular programming language in the world
* It has a large demand in the current job market
* It is easy to learn and simple to use
* It is open-source and free
* It is secure, fast and powerful
* It has a huge community support (tens of millions of developers)
* Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs

## Java Install

Some PCs might have Java already installed.

To check if you have Java installed on a Windows PC, search in the start bar for Java or type the following in Command Prompt (cmd.exe):

C:\Users\Your Name>java -version

If Java is installed, you will see something like this (depending on version):

java version "11.0.1" 2018-10-16 LTS  
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)  
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)

If you do not have Java installed on your computer, you can download it for free at [oracle.com](https://www.oracle.com/technetwork/java/javase/overview/index.html).

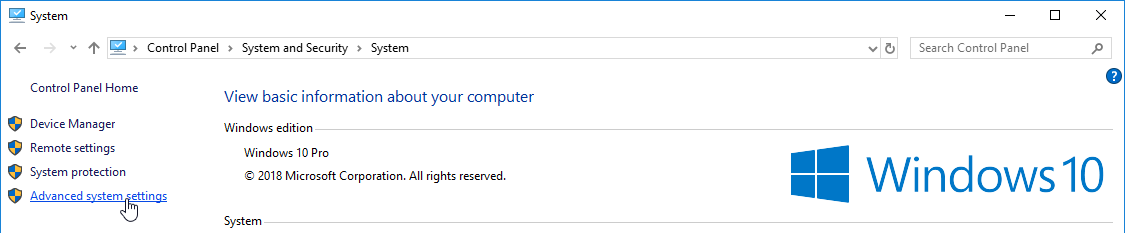
Setup for Windows

To install Java on Windows:

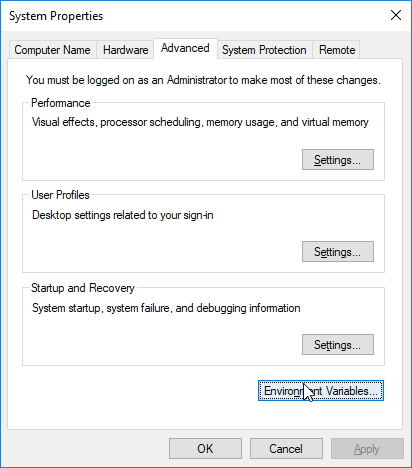
1. Go to "System Properties" (Can be found on Control Panel > System and Security > System > Advanced System Settings)
2. Click on the "Environment variables" button under the "Advanced" tab
3. Then, select the "Path" variable in System variables and click on the "Edit" button
4. Click on the "New" button and add the path where Java is installed, followed by **\bin**. By default, Java is installed in C:\Program Files\Java\jdk-11.0.1 (If nothing else was specified when you installed it). In that case, You will have to add a new path with: **C:\Program Files\Java\jdk-11.0.1\bin**  
   Then, click "OK", and save the settings
5. At last, open Command Prompt (cmd.exe) and type **java -version** to see if Java is running on your machine

Show how to install Java step-by-step with images »

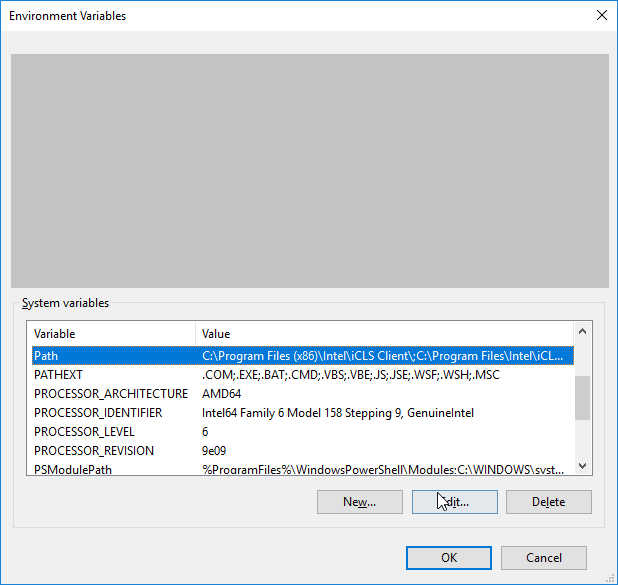
Step 1

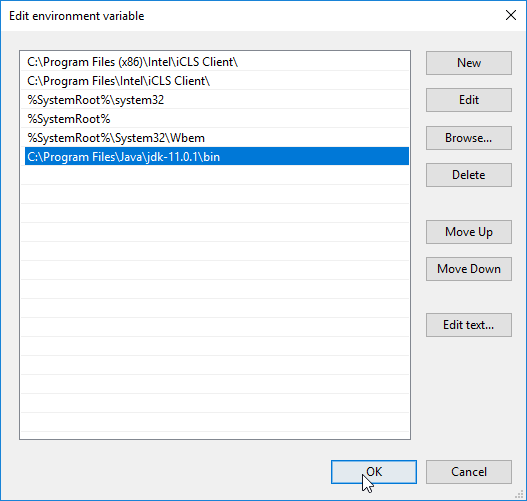


Step 2



Step 3



Step 4

Step 5

Write the following in the command line (cmd.exe):

C:\Users\*Your Name*>java -version

If Java was successfully installed, you will see something like this (depending on version):

java version "11.0.1" 2018-10-16 LTS  
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)  
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)

## Java Quick start

In Java, every application begins with a class name, and that class must match the filename.

Let's create our first Java file, called Main.java, which can be done in any text editor (like Notepad).

The file should contain a "Hello World" message, which is written with the following code:

Main.java

public class Main {

public static void main(String[] args) {

System.out.println("Hello World");

}

}

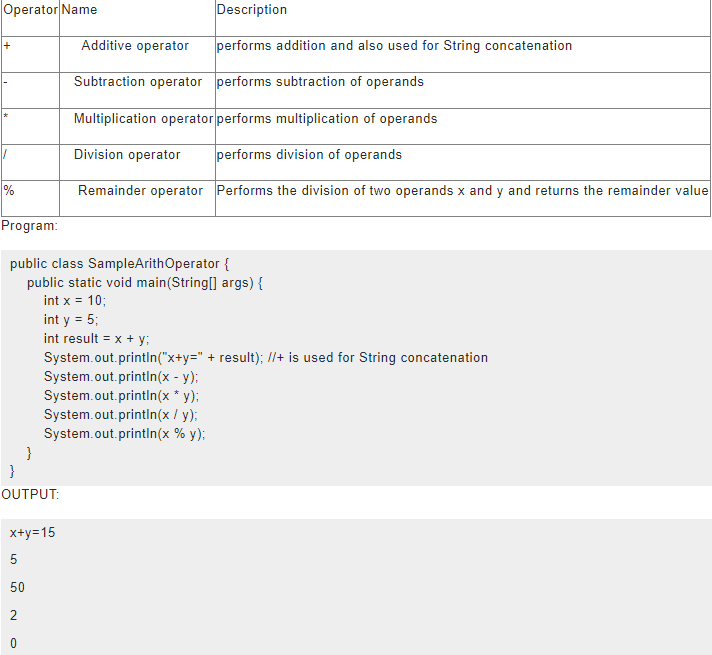
# Operators

Operators are nothing but the symbols that perform specific operations. In this chapter let us discuss in detail the different types of operators Basic terminologies which we are going to use often are operator and operand If X+Y is the operation X and Y are the operands,+ is the operator

The following section will describe the types of operators

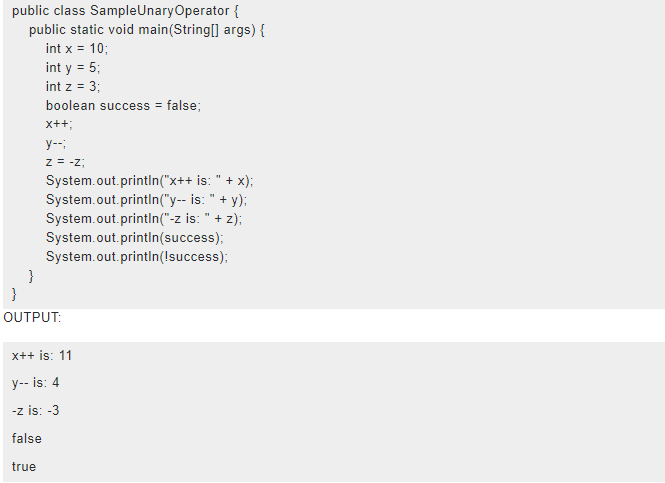
**1. Arithmetic Operators**

Arithmetic operators perform mathematical functions like addition, subtraction, multiplication, division, and modulor division. It requires a minimum of two operands.



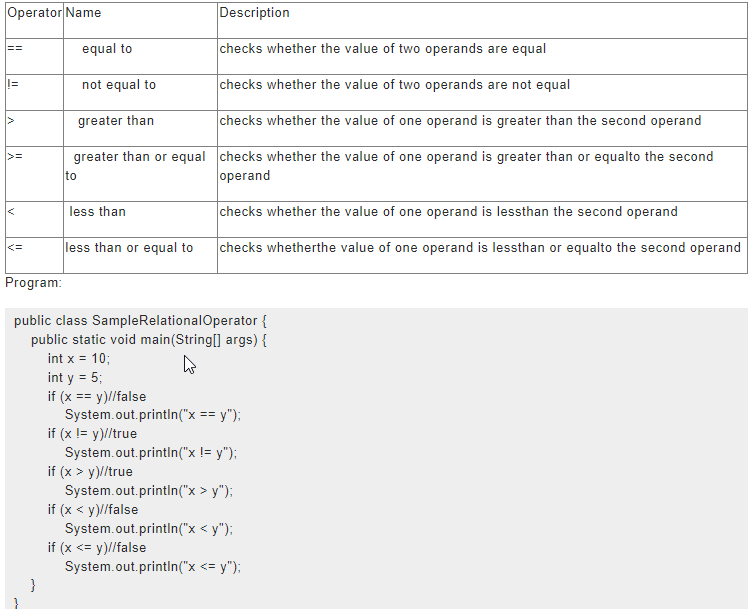
**2. Unary Operators**

The unary operators perform various operations such as incrementing or decrementing a value by one, negating an expression, or inverting the value of a boolean. Unary operator requires only one operand.



**3. The Equality and Relational Operators:**

The equality and relational operators are used to determine whether one operand is greater than or less than or equal to or not equal to another operand.Here equal to is represented as == not =

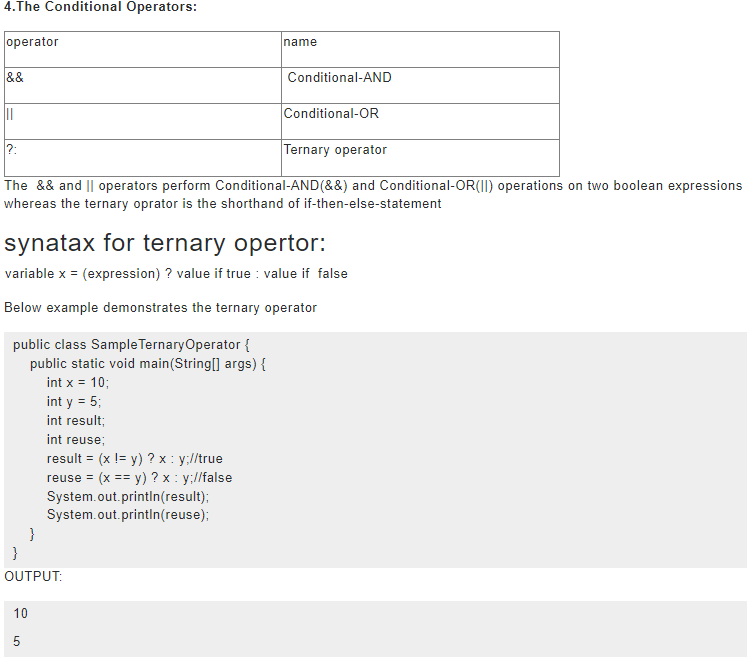


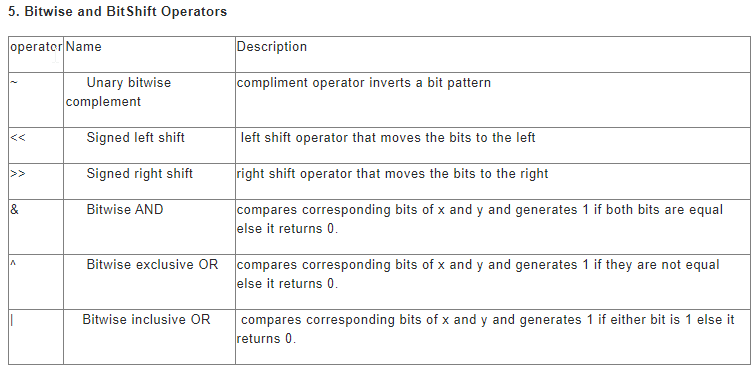
In the above sample, only the true statements will be executed.

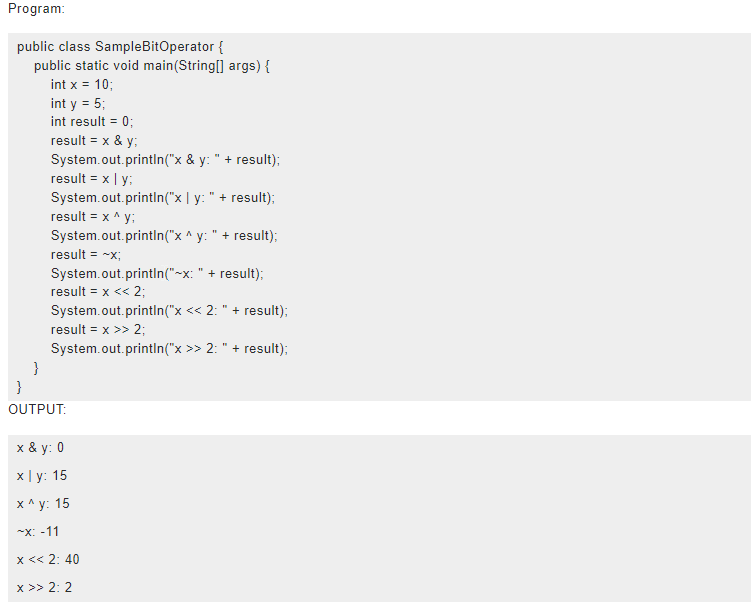
OUTPUT:

x != y

x > y







# Variables

A Variable is a name holding a value.Java programming language is **statically-typed** which means the variable needs to be declared before assigning value to it and the value of the variable can be changed during execution of the program based on the operations performed in the program

While declaring a variable a particular memory space is created for that variable based on the data type of the variable

**Syntax to declare a variable in java**

A variable can be declared as below



Java has the option to declare a variable without value and the value can be assigned during execution of the program



**Types of variables**

The different types of variables in java are as follows

**Static(class)variables**

* Static variables are also known as class variables
* It is declared with the keyword “static”
* It holds the same value regardless of how many times the class is instantiated
* Memory allocation for the static variables happens only once during the class loading

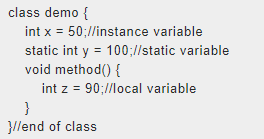
**Instance Variables (Non-Static)**

* Instance variables are also known as Non static variables
* The value of instance variable is unique to each instance of the class

**Local** **Variables**:

* Local variables  are temporary setup to access a variables inside a method
* It doesnot have any particular keyword it depends on where it is declared,local variables should be declared inside the opening and closing braces of a method
* Local variables are available only inside the method it is not available for the rest of the class

Sample program to understand variables:



**The naming convention for variables**

The set of rules to be considered while naming a variable is as follows

* Variable names are case-sensitive.
* White space is not permitted while naming a variable
* Variable name should not be a Keyword or reserved word
* While naming a variable it is highly recommended to use the full name instead of abbreviations this helps in easy understanding of the code
* It is highly recommended to use a letter as the starting letter of a variable. Usage of $ and \_ is not recommended
* If the variable name is one-word use lowercase, If its two words the first word should be in lowercase and the starting letter of the second word should be in uppercase (Ex: power calculation)

# Data type

### 

The Java programming language is **statically-typed**, the variable should be initialized in the program before using it.

The data type of the variable determines the type of value the variable is allowed to hold

**Primitive Data type:**

A primitive type is predefined by the language and is named by a reserved keyword. Primitive values do not share state with other primitive values.

The **eight primitive data types** supported by the Java programming language are:

**1.byte**

* The byte data type is an 8-bit signed two's complement integer
* It has a minimum value of -128 and a maximum value of 127 (inclusive)
* The byte data type can be useful for saving memory in large arrays, where the memory savings actually matters
* The default value of byte is 0
* **Example:** byte a = 126

**2. short:**

* The short data type is a 16-bit signed two's complement integer.
* It has a minimum value of -32,768 and a maximum value of 32,767 (inclusive)
* Like byte you can use a short to save memory in large arrays, in situations where the memory savings actually matters
* The default value of short is 0
* **Example:** short = 56;

**3. int:**

* int data type is a 32-bit signed two's complement integer
* It has a minimum value of -231 and a maximum value of 231-1
* In Java SE 8 and later, you can use the int data type to represent an unsigned 32-bit integer, which has a minimum value of 0 and a maximum value of 232-1.
* The default value is 0
* **Example:**int i=89;

**4. Long:**

* The long data type is a 64-bit two's complement integer.
* The signed long has a minimum value of -263 and a maximum value of 263-1
* In Java SE 8 and later, you can use the long data type to represent an unsigned 64-bit long, which has a minimum value of 0 and a maximum value of 264-1
* Default value is 0L
* **Example:** long a = 100000L

**5. float:**

* The float data type is a single-precision 32-bit IEEE 754 floating point
* float can be used instead of double if you need to save memory in large arrays of floating-point numbers
* It should not be used for precise values, such as currency.
* The default value is 0.0f
* **Example:** float f = 4.7333434f;

**6. double:**

* The double data type is a double-precision 64-bit IEEE 754 floating point.
* It is highly recommended to store decimal values
* Like float, this data type should never be used for precise values, such as currency
* Default value is 0.0d

**Example:** double d = 4.355453532

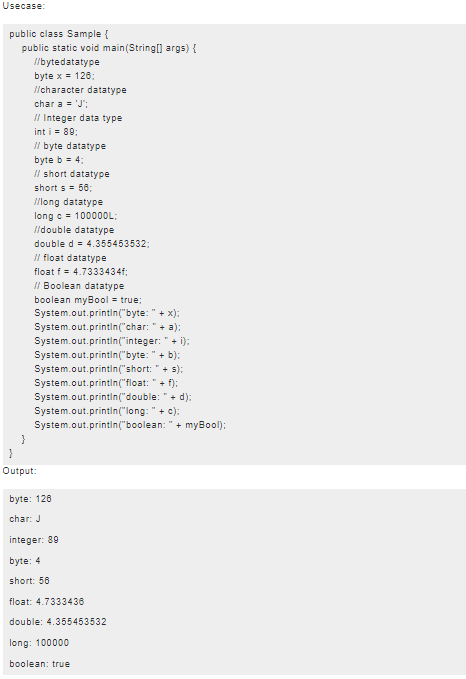
**7. Boolean:**

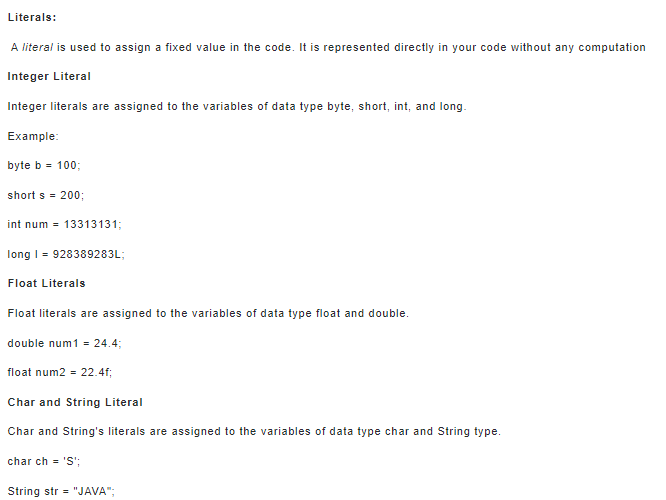
* boolean data type represents one bit of information
* The boolean data type has only two possible values: true and false
* It is used to track simple flag true/false conditions
* Default value is false

**Example:** boolean myBool = true;

**8. char:**

* The char data type is a single 16-bit Unicode character
* It has a minimum value of '\u0000' (or 0) and a maximum value of '\uffff' (or 65,535 inclusive)
* Char data type is used to store any character
* **Example:** char myText= 'D';





# Exception handling

### 

This is the most significant feature of java

An exception is defined as an event, which occurs during the execution of a program that disrupts the normal flow of the program's instructions.

Whenever an exception occurs a runtime time object is created which is called **exception object** which contains information like type and the state of the program when the error occurred

Creating an exception object and handing it to the runtime system is called throwing an exception

Whenever an exception occurs a runtime time object searches for a block of code to handle it which is called **exception handler**which will catch the exception

Exceptions are used to indicate many different types of error conditions which can be categorized as follows.

**JVM Errors:**

* OutOfMemoryError
* StackOverflowError
* LinkageError

**System errors:**

* FileNotFoundException
* IOException
* SocketTimeoutException

**Programming errors:**

* NullPointerException
* ArrayIndexOutOfBoundsException
* ArithmeticException

**Advantages of exception handling**

* Exceptions separate error-handling code from regular code
* Exceptions propagate errors up the call stack.
* Exception classes group and differentiate error types.
* Exceptions standardize error handling.

#### ****Keywords in java exception****

**try**

try statement should be specified in the block where the exception needs to be handled.It is followed by catch or finally which means try cannot be used alone

**catch**

catch block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.

**Finally**

Finally block is used to execute the important code of the program. It is executed whether an exception is handled or not.

**Throw**

Throw keyword is used to throw an exception.

#### ****Types of exception****

**Checked exception**

The classes which directly inherit the Throwable class except RuntimeException and Error are known as checked exceptions e.g. IOException, SQLException, etc. Checked exceptions are checked at compile-time so this needs to handle which can be just logging the exception or returning error message to the user

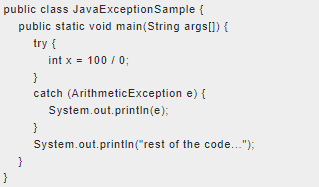
**Unchecked exception**

The classes which inherit RuntimeException are known as unchecked exceptions e.g. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time, but they are checked at runtime and this can be ignored sometimes

**Error**

Error is irrecoverable e.g. OutOfMemoryError, VirtualMachineError, AssertionError etc.

Simple use case



Hints to understand the basic exception categories

**NullPointer Exception**

The parameter value is null while performing an operation

**ArithmeticException**

If we perform operations like division by zero

**NumberFormatException**

This occurs when a string value is assigned to an integer datatype variable

**ArrayIndexOutOfBoundsException**

This occurs when you are inserting any value in the wrong index

**IllegalArgumentException**

Non-null parameter value is inappropriate

**IllegalStateException**

Object state is inappropriate for method invocation

**IndexOutOfBoundsException**

Index parameter is out of range

**UnsupportedOperationException**

Object does not support the method