**Usman Institute of Technology**

**Department of Computer Science**

**Course Code: SE312**

**Course Title: Software Construction and Development**

**SPRING 2024**

**Lab 04**

**Objective: Understand the basic syntax of Java, including data types, variables, operators, and control flow structures.**

**Student Information**

|  |  |
| --- | --- |
| Student Name |  |
| Student ID |  |
| Date |  |

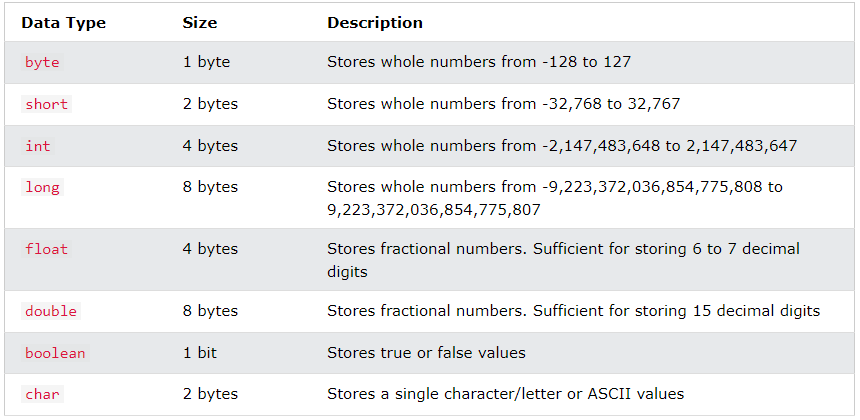
**Assessment**

|  |  |
| --- | --- |
| Marks Obtained |  |
| Remarks |  |
| Signature |  |

**LAB #04**

**Data types**

* In programming, a data type is a classification that specifies which type of value a variable can hold and what operations can be performed on that variable.
* Data types define the structure and behavior of variables in a programming language, allowing the compiler or interpreter to allocate memory and perform operations efficiently.
* **Primitive data types**
* Java has eight primitive data types: byte, short, int, long, float, double, char, and boolean.
* Primitive data types represent single values and are not objects.
* They have predefined sizes and ranges.



* **Reference Data Types**

**Example 1: Using String reference data type**

String message = "Hello, Java!";

System.out.println(message);

**Example 2: Using arrays (reference data type)**

int[] numbers = {1, 2, 3, 4, 5}; // Declaration and initialization of an integer array

System.out.println("First element: " + numbers[0]);

**Example 3: Manipulating string objects**

String firstName = "Sitwat";

String lastName = "Ashraf";

String fullName = firstName + " " + lastName;

System.out.println("Full Name: " + fullName);

**Variables**

* Variables are used to store data values in Java.
* Variable names must start with a letter, underscore, or dollar sign, followed by letters, digits, underscores, or dollar signs.

**Declaring (Creating) Variables**

To create a variable, you must specify the type and assign it a value:

type variableName = value;

**Example**

int x = 5;

double y = 4.5;

**\*\*Lab Activity**

1. Create a variable called name of type String and assign it the value "your name".
2. Create a variable that should store your Roll number.

You can also declare a variable without assigning the value, and assign the value later:

int myRollNum;

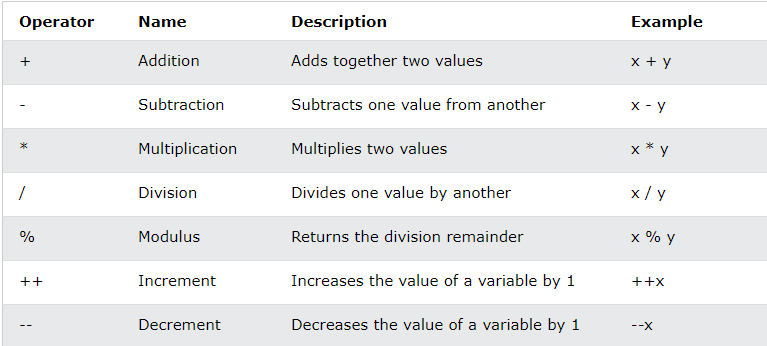
myRollNum = 001;

System.out.println(myRollNum);

**Operators**

* Operators are used to perform operations on variables and values.
* Java divides the operators into the following groups:
* **Arithmetic operators**

Arithmetic operators are used to perform common mathematical operations.

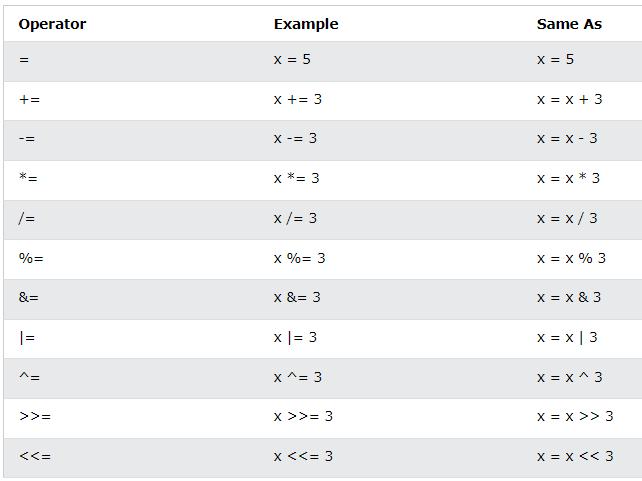


* **Assignment Operators**

Assignment operators are used to assign values to variables.

**Example**

int x = 10;



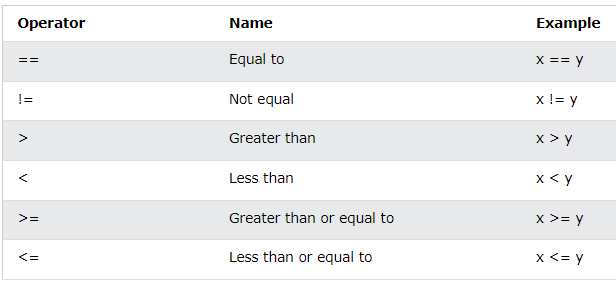
* **Comparison Operators**
* Comparison operators are used to compare two values (or variables).
* This is important in programming, because it helps us to find answers and make decisions.
* The return value of a comparison is either true or false

**Example**

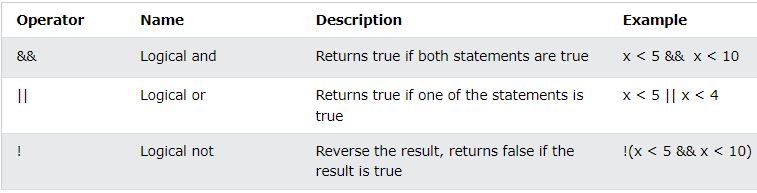
int x = 5;

int y = 3;

System.out.println(x > y);



* **Logical Operators**
* You can also test for true or false values with logical operators.
* Logical operators are used to determine the logic between variables or values.



**Control flow structures**

* **If-Else Statement:**

The if-else statement executes a block of code if a specified condition is true. If the condition is false, another block of code can be executed.

**Example**

int number = 10;

if (number > 0) {

System.out.println("Number is positive");

} else {

System.out.println("Number is negative");

}

* **Switch Statements**
* Instead of writing many if..else statements, you can use the switch statement.
* The switch statement selects one of many code blocks to be executed.
* The switch statement works with;
* The switch expression is evaluated once.
* The value of the expression is compared with the values of each case.
* If there is a match, the associated block of code is executed.
* The break and default keywords are optional.

**Example**

int day = 4;

switch (day) {

case 1:

System.out.println("Monday");

break;

case 2:

System.out.println("Tuesday");

break;

case 3:

System.out.println("Wednesday");

break;

case 4:

System.out.println("Thursday");

break;

case 5:

System.out.println("Friday");

break;

case 6:

System.out.println("Saturday");

break;

case 7:

System.out.println("Sunday");

break;

}

**Break**

In Java, when the program encounters a break keyword within a switch block, it immediately exits the switch block. This termination prevents further execution of code within the block and halts any additional case testing. Once a match is found and the corresponding actions are performed, the break statement signals that further evaluation is unnecessary, thus ending the switch block's execution at that point.

**Default**

The default keyword specifies some code to run if there is no case match.

* **Loops**

Loops are capable of repeatedly executing a block of code until a certain condition is met. They offer practical benefits such as time-saving, error reduction, and improved code readability.

* **While loop**

The while loop iterates over a block of code as long as a given condition remains true:

**Example**

public class lab4 {

public static void main(String[] args) {

int i = 0;

while (i < 5) {

System.out.println(i);

i++;

}

}

}

* **Do/while loop**

The do/while loop serves as an alternative to the while loop. It begins by executing the code block once, then evaluates the condition. If the condition is true, it continues to repeat the loop; otherwise, it terminates.

**Example**

public class lab4 {

public static void main(String[] args) {

int i = 0;

do {

System.out.println(i);

i++;

}

while (i < 5);

}

}

* **For loop**

When you have a predetermined number of iterations for looping through a code block, it's preferable to use the for loop over a while loop.

**Example**

for (int i = 0; i < 5; i++) {

System.out.println(i);

}

**TASKS**

1. Write a Java program that prompts the user to enter marks obtained in three subjects and calculates the total marks. Display the total marks obtained by the student.
2. Write a Java program that prompts the user to enter ages of three people and determines the oldest person among them. Display the age of the oldest person.
3. Write a Java program to check whether a person is eligible to vote. Prompt the user to enter their age and display whether they are eligible to vote or not.
4. Write a Java program to print the multiplication table of a given number (n). Prompt the user to enter a number (n) and then print its multiplication table up to 10.
5. Write a Java program to calculate the sum of numbers from 1 to a given number (n). Prompt the user to enter a number (n) and then calculate the sum of all numbers from 1 to n. Display the result.

**How to Submit**

* Submit lab work in a single pdf/docx on MS Team.
* Submit the work as per format given in this manual (No other format will be accepted).
* Lab work (Exercises) file name should be saved with your roll number and course code (e.g. 21B-001-SE\_SExxx\_LWxx.pdf where SExxx is course code and LWxx is Lab number).