DAY 1 and 2

1 Primitive Data Types

Ans JAVA has 8 primitive data types. They are byte, short, int, long, float, double, char, Boolean.

1. Difference between primitive and non primitive data types

Ans Primitive are user defined datatype and non primitive are user defined

4 **Type casting** in Java is the process of converting a variable from one data type to another.

**1. Implicit Casting (Widening)**

Done automatically by Java when converting a smaller type to a larger type.

java

int num = 10;

double result = num; // Implicit casting from int to double

System.out.println(result); // 10.0

**2. Explicit Casting (Narrowing)**

Done manually using parentheses.

java

double value = 9.78;

int result = (int) value; // Explicit casting from double to int

System.out.println(result); // 9

5 the **default value** for each primitive data type in Java:

| **Data Type** | **Default Value** |
| --- | --- |
| **byte** | 0 |
| **short** | 0 |
| **int** | 0 |
| **long** | 0L |
| **float** | 0.0f |
| **double** | 0.0d |
| **char** | null character |
| **boolean** | false |

**Types of Control Statements in Java**

**1. Decision-Making Statements**

* Used to execute code blocks based on conditions.

**a) if statement**

int num = 10;

if (num > 5) {

System.out.println("Number is greater than 5");

}

**b) if-else statement**

int num = 3;

if (num > 5) {

System.out.println("Greater");

} else {

System.out.println("Smaller");

}

**c) if-else-if ladder**

int marks = 85;

if (marks >= 90) {

System.out.println("Grade A");

} else if (marks >= 75) {

System.out.println("Grade B");

} else {

System.out.println("Grade C");

}

**d) switch statement**

int day = 3;

switch(day) {

case 1: System.out.println("Monday"); break;

case 2: System.out.println("Tuesday"); break;

case 3: System.out.println("Wednesday"); break;

default: System.out.println("Invalid day");

}

**2. Looping Statements**

* Used to execute code repeatedly.

1. **for loop**
2. for (int i = 1; i <= 5; i++) {

System.out.println(i);

}

**b) while loop**

int i = 1;

while (i <= 5) {

System.out.println(i);

i++;

}

**c) do-while loop**

int i = 1;

do {

System.out.println(i);

i++;

} while (i <= 5);

**3. Jump Statements**

* Used to change the normal flow of execution.

**a) break**

for (int i = 1; i <= 5; i++) {

if (i == 3) break;

System.out.println(i);

}

**b) continue**

for (int i = 1; i <= 5; i++) {

if (i == 3) continue;

System.out.println(i);

}

**c) return**

public static int add(int a, int b) {

return a + b;

}

Difference between break and skip

The **break** statement terminates the loop entirely, while the **continue** statement skips the current iteration and moves to the next one.

A **while** loop checks the condition before executing the block, whereas a **do-while** loop checks the condition after executing the block at least once.

**10 keywords** in Java:

class, public, static, void, if, else, switch, break, continue, return

**static**: Used to define members (variables, methods, blocks) that belong to the class rather than an instance.

**final** Used to declare constants, prevent method overriding, and prevent class inheritance.

**this** Refers to the current object of the class.

**super** Refers to the parent class object and can be used to access parent methods, variables, or constructors.

The main **types of operators** in Java are:

1. **Arithmetic Operators** : +, -, \*, /, %
2. **Relational Operators** : ==, !=, >, <, >=, <=
3. **Logical Operators** : &&, ||, !
4. **Assignment Operators** : =, +=, -=, \*=, /=, %=

| **Data Type** | **Size (bits)** | **Range** |
| --- | --- | --- |
| **byte** | 8 | -128 to 127 |
| **short** | 16 | -32,768 to 32,767 |
| **int** | 32 | -2,147,483,648 to 2,147,483,647 |
| **long** | 64 | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| **float** | 32 | ~±3.4 × 10³⁸ (6–7 decimal digits) |
| **double** | 64 | ~±1.8 × 10³⁰⁸ (15–16 decimal digits) |
| **char** | 16 | 0 to 65,535 (Unicode characters) |
| **boolean** | JVM-dependent (1 bit conceptually) | true or false |

In Java, **char** represents a single character, while **String** represents a sequence of characters enclosed in double quotes.

**wrapper classes** are object representations of primitive data types

The **instance of** keyword in Java is used to check whether an object is an instance of a specific class or implements a specific interface.

**==** : Compares **references** of objects,

**equals()** :Compares the **contents/values** of objects

**this** : Refers to the current class’s instance, and can be used to call the **overridden method** of the current class.

**super** : Refers to the parent class’s instance, and is used to call the **overridden method** of the superclass.

Bitwise operators in Java are:

&, |, ^, ~, <<, >>, >>>