Java Data Type

Java 17 supports the same data types as earlier versions of Java, divided into **primitive types** and **reference types**. Below is a detailed overview:

---

**Primitive Data Types**

These are the basic data types in Java:

1. **Integer Types**
    - **byte**: 8-bit, values from -128 to 127
    - **short**: 16-bit, values from -32,768 to 32,767
    - **int**: 32-bit, values from $-2^{31}$ to $2^{31}-1$
    - **long**: 64-bit, values from $-2^{63}$ to $2^{63}-1$ (add L for literals, e.g., 12345L)
2. **Floating-Point Types**
    - **float**: 32-bit, single-precision
    - **double**: 64-bit, double-precision (default for decimal values)
3. **Character Type**
    - **char**: 16-bit, represents a single Unicode character (e.g., 'a', '@')
4. **Boolean Type**
    - **boolean**: Only two values: true or false

---

**Reference Data Types**

These are used to store objects and arrays.

1. **Classes** Example: String, Integer, Double, or user-defined classes.
2. **Interfaces** Example: Runnable, List, Map.
3. **Arrays** Example: int[], String[].
4. **Enums** Used to define a set of constants. Example:

enum Day { MONDAY, TUESDAY, WEDNESDAY };

var (Local Variable Type Inference) Introduced in Java 10, allows inference of data types at compile time.
Example:
var message = "Hello, Java 17"; // `message` is inferred as String

**Type Conversion**

**Automatic type conversion**

**Automatic type conversion happens when:**

- **The target type is larger than the source type.**
- **No explicit casting is required.**

**Common conversions:**

- **byte → short → int → long → float → double**
- **char → int**

Conversion examples

Example 1: Converting int to long

```
public class TypeConversionExample {
   public static void main(String[] args) {
      int num = 100; // int type
      long bigNum = num; // Automatic conversion to long
      System.out.println("Integer value: " + num);
      System.out.println("Long value: " + bigNum);
   }
}
```

Example 2: Converting float to double

```
public class TypeConversionExample {
   public static void main(String[] args) {
      float decimal = 12.34f; // float type
      double biggerDecimal = decimal; // Automatic conversion to double
      System.out.println("Float value: " + decimal);
      System.out.println("Double value: " + biggerDecimal);
   }
}
```

Example 3: Mixed-Type Arithmetic Operations

```
public class TypeConversionExample {
   public static void main(String[] args) {
      int intNum = 50;
      double doubleNum = 5.5;

      // int is automatically converted to double during addition
      double result = intNum + doubleNum;
      System.out.println("Result of addition: " + result);
   }
}
```

Example 4 Type Conversion Example

```
public class TypeConversionExample {
   public static void main(String[] args) {
      char letter = 'A'; // char type
      int asciiValue = letter; // Automatic conversion to int
      System.out.println("Character: " + letter);
```

```
        System.out.println("ASCII value: " + asciiValue);
    }
}
```

Example 5: Expression Evaluation

```
public class TypeConversionExample {
    public static void main(String[] args) {
        byte a = 10;
        byte b = 20;
        // a and b are promoted to int during addition
        int sum = a + b;
        System.out.println("Sum: " + sum);
    }
}
```

Check your understanding

Question 1: Widening Conversion

```
public class TypeConversionQuiz {
    public static void main(String[] args) {
        byte b = 10;
        int i = b;
        double d = i;
        System.out.println("Byte to int: " + i);
        System.out.println("Int to double: " + d);
    }
}
```

Ans

---

Question 2: Narrowing Conversion

```
public class TypeConversionQuiz {
    public static void main(String[] args) {
        double d = 100.04;
        int i = (int) d; // Explicit narrowing
        System.out.println("Double to int: " + i);
    }
}
```

Ans

---

Question 3: Char to Int

```
public class TypeConversionQuiz {
    public static void main(String[] args) {
        char c = 'A';
        int i = c;
        System.out.println("Char to int: " + i);
    }
}
```

Ans

---

Question 4: Expression Evaluation

```
public class TypeConversionQuiz {
    public static void main(String[] args) {
        byte b = 42;
        byte c = 20;
        byte result = (byte) (b + c); // Explicit cast needed
        System.out.println("Result: " + result);
    }
```

```
}
```
Ans

---

Question 5: Mixed-Type Arithmetic

```
public class TypeConversionQuiz {
   public static void main(String[] args) {
      int i = 10;
      float f = 3.5f;
      double result = i * f;
      System.out.println("Result: " + result);
   }
}
```
Ans

---

Question 6: Float to Int

```
public class TypeConversionQuiz {
   public static void main(String[] args) {
      float f = 10.99f;
      int i = (int) f;
      System.out.println("Float to int: " + i);
   }
}
```
Ans

---

Question 7: Default Type in Decimal Values

```
public class TypeConversionQuiz {
   public static void main(String[] args) {
      double d = 10.5; // Implicitly double
      float f = 10.5f; // Explicit float
      System.out.println("Double: " + d);
      System.out.println("Float: " + f);
   }
}
```
Ans

---

Question 8: Implicit vs Explicit Casting

```
public class TypeConversionQuiz {
   public static void main(String[] args) {
      long l = 100000L;
      int i = (int) l; // Explicit cast
      short s = (short) i; // Explicit cast
      System.out.println("Long to int: " + i);
      System.out.println("Int to short: " + s);
   }
}
```
Ans

---

Question 9: Arithmetic Promotion

```
public class TypeConversionQuiz {
   public static void main(String[] args) {
      short s = 5;
      int i = 10;
      double d = 20.0;
      double result = s + i * d; // Promotion to double
      System.out.println("Result: " + result);
   }
}
```

Ans

---

Question 10: Char and Byte Arithmetic

```
public class TypeConversionQuiz {
   public static void main(String[] args) {
      char c = 'A'; // ASCII value 65
      byte b = 5;
      int result = c + b; // Promoted to int
      System.out.println("Result: " + result);
   }
}
```

Ans