

Data types, Variables and Array

❖ Data type in Java:-

Data types specify the types of value and the different sizes that can be stored in variable. Java is a strongly typed language. It means , all variables must before its use.

- **Types of data types**

1. Primitive data types
2. Non-primitive data types
- 3.

1. Primitive data types:-

Primitive types are the most basic data types available in the java language. Primitive types are predefined. These are also commonly referred to as simple types.

- **Java defines eight primitive types of data:-**

byte	short	
int	boolean	char
float	double	long

- **Integers:-**

An integers is whole number that can be positive, negative or zero .

Java defines four integer types: byte , short, int, long.

byte-

The smallest integer type is byte. This is signed 8-bit type that has a range from -128 to 127. It is useful when we are working with a stream of data from a network or file.

- Its default value is 0.
- Size 1byte(8 bit).
- The byte data type use to save memory in large arrays.
- Its saves space because a byte is 4 times smaller than a integer.
- Byte variables are declared by use of the **byte** keyword.

- **Syntax:-**

```
byte bytevar;
```

For example

```
byte b, c;
```

short-

short is a signed 16-bit type. It has a range from -32,768 to 32,767. It is probably the least-used Java type.

- Its default value is 0.
- Size 2 byte(16 – bits)
- It can also save memory just like byte data type.

- Short variables are declared by use of the **short** keyword.
- **Syntax:-**
short shortvar;

For example
short s;

int-

The most commonly used integer type is **int**. It is a signed 32-bit type that has a range from $-2,147,483,648$ to $2,147,483,647$.

- Its default value is 0.
- Size 4 byte (32 bits)
- The int data type generally used as a default data type for integral values.
- **Syntax:-**
int intvar;

For example
int a;

long-

long is a signed 64-bit type and is useful for those occasions where an **int** type is not large enough to hold the desired

value. The range of a **long** is quite large. This makes it useful when big, whole numbers are needed.

- Its default value is 0.
- Size 8 byte(64 bits)
- Its useful to store large range of value.
- **Syntax:-**
long longvar;

For example
long a = 40000L;

➤ **Float types :-**

Floating-point numbers, also known as *real* numbers, are used when evaluating expressions that require fractional precision.

Name	Width in Bits	Approximate Range
Double	64	4.9e-324 to 1.8e+308
Float	32	1.4e-045 to 3.4e+038

float-

The type **float** specifies a *single-precision* value that uses 32 bits of storage.

- Single precision is faster on some processors and takes half as much space as double precision.
- Its default value is 0.0F.
- Size 4 byte (32 bits)
- Its value range is unlimited.
- End the value with an "f".

- **Syntax:-**

float floatvar;

For example
float f = 3.14f

double-

Double precision, as denoted by the **double** keyword, uses 64 bits to store a value. Double precision is actually faster than single precision.

- Its default value is 0.0d.
- Size 8 byte (64 bits)
- Its value range is unlimited.
- End the value with an "d".

- **Syntax:-**

double doublevar;

For example
double a = 23.4d

➤ **Characters :-**

The data type used to store characters is **char**. The char data types is a single 16 – bit Unicode character. The range of a **char** is 0 to 65,536. There are no negative **chars**. The standard set of characters known as ASCII still ranges from 0 to 127 as always, and the extended 8-bit character set, ISO-Latin-1, ranges from 0 to 255.

- Its default value is ‘\u000’.
- Size 2 byte (16 bits)
- **char** in Java is not the same as **char** in C or C++.
- **Syntax:-**
char charvar;

For example
char c ;

➤ **Boolean :-**

Java has a primitive type, called **boolean**, for logical values. It can have only one of two possible values, **true** or **false**. This data type is used for simple flags that track true/false conditions. It specifies one bit of information.

- Its default value is false.
- Size virtual machine dependent.
- **Syntax:-**
boolean boolvar;

For example:-
boolean b = true;

2. Non primitive data types:-

Non primitive data types are created by programmers. They are not predefined in java like primitive data types. When we define a Variable of non-primitive data types , it references a memory location where data is stored in the heap memory . i.e it reference to a memory where an object is actually placed. So it is also called reference data types or object reference variable.

The non – primitive data types includes

String Arrays
Classes interfaces etc..

➤ **String:-**

Strings are defined as an array of characters. The difference between a character array and a string is the string is terminated with a special character '\0'.

- **Syntax:-**
String stvar;

For Example
String st;

Variables

❖ Variable in Java:-

Variable is the name of memory place where we can store specific type of data within specific range. Variable may change their value during program execution. The variable is the basic unit of storage in a Java program. A variable is defined by the combination of an identifier, a type, and an optional initializer. In addition, all variables have a scope, which defines their visibility, and a lifetime.

- The Rules for Constructing variable Name.
 - Names can contain letters, digits, underscores, and dollar signs.
 - Names must begin with a letter.
 - Names should start with a lowercase letter and it cannot contain whitespace.
 - Names can also begin with \$ and _ (but we will not use it in this tutorial).
 - Names are case sensitive ("myVar" and "myvar" are different variables).
 - Reserved words (like Java keywords, such as int or boolean) cannot be used as names

- Declaration of variable and initialization –
datatype identifier [= value][, identifier [= value] ...] ;

The datatype is one of Java's atomic types. The *identifier* is the name of the variable. we can initialize the variable by specifying an equal sign and a value. To declare more than

one variable of the specified type, use a comma separated list.

Ex:-

```
int a,b,c=5;
```

```
char c;
```

```
String st="BCA";
```

```
Boolean b;
```

- **Dynamic initialization –**

Java allows variables to be initialized dynamically, using any expression valid at the time the variable is declared.

Ex:-

```
int a =5;
```

```
float b = a*0.5; //dynamically initialized
```