

01. STATICALLY TYPED PROGRAMMING LANGUAGES:

In statically typed programming languages, the programmer must explicitly define the data type of each variable, and the compiler checks the compatibility of these variables with their declared types. This helps catch type-related errors at the compile-time.

DYNAMICALLY TYPED PROGRAMMING LANGUAGES:

A dynamically typed programming language is a type of programming language in which variable types are determined at runtime, rather than being explicitly defined during the code compilation or declaration phase. In dynamically typed languages, the type of a variable can change during the execution of the program.

STRONGLY TYPED LANGUAGES:

A strongly typed programming language is a type of programming language in which the data type of a variable is explicitly defined and strictly enforced. In such languages, once a variable is declared with a specific data type, it cannot be implicitly or automatically converted to another type without explicit type casting.

LOOSELY TYPED LANGUAGE:

A loosely typed programming language is a type of programming language where variables are not explicitly bound to a specific data type during their declaration. In other words, in a loosely typed language, the data type of a variable can change dynamically at runtime based on the value assigned to it.

Java is a statically typed, dynamically typed and strongly typed programming language.

02. CASE SENSITIVE:

When a programming language is case sensitive, it means that it distinguishes between uppercase and lowercase letters in identifiers. This means that "MyNumber," "mynumber," and "MYNUMBER" are all considered different identifiers.

CASE INSENSITIVE:

When a programming language is case insensitive, it means that it does not distinguish between uppercase and lowercase letters in identifiers. This means that "MyNumber" "mynumber," and "MYNUMBER" are all considered the same identifier.

CASE SENSITIVE-INSENSITIVE:

Sometimes, a programming language can be both case sensitive and case insensitive, depending on the context. This usually occurs when the language treats certain aspects as case insensitive, while other parts remain case sensitive. For instance, keywords and variable names might be case insensitive, while function names are case sensitive.

Classification of Java:

Java is a case-sensitive programming language. It follows the general convention in most programming languages where identifiers are case sensitive. This means that "myVariable" and "myvariable" are considered different identifiers in Java.

03. IDENTITY CONVERSION

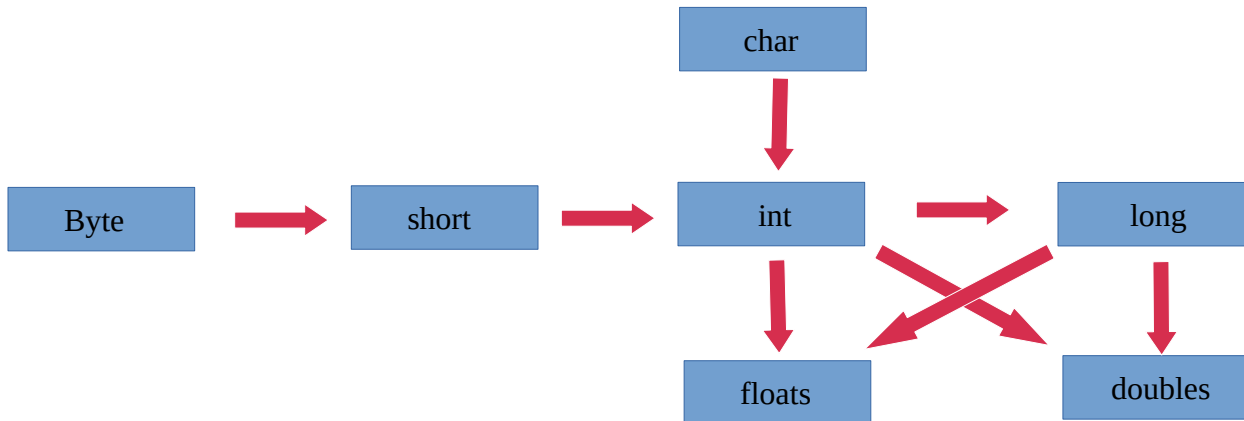
In Java, the concept of Identity Conversion refers to the conversion of a value from one type to the same type. In identity Conversion, no actual conversion is performed on the value itself.

```
int intNumber = 25;  
long longNumber = intNumber;
```

```
short shortNum = 123;  
int intNum = shortNum;
```

04. Primitive widening conversion is an automatic and implicit conversion that takes place when a value of a smaller data type is assigned to a variable of a larger data type.

```
char myChar = '5';  
int myInt = myChar;
```



05. COMPILE-TIME CONSTANTS:

Compile-time constants are constants whose values are known by the compiler and can be determined at compile time.

```
final int MY_INT = 10;
```

RUNTIME CONSTANTS:

Runtime constants are constants whose values are determined and assigned during the program's execution, i.e., at runtime. These constants are not known to the compiler during the compilation process

```
final int MY_INT = 25*(int)Math.random();
```

06. IMPLICIT NARROWING PRIMITIVE CONVERSIONS:

This also known as automatic conversions, occur when a value of a larger data type is assigned to a variable of a smaller data type except doubles, floats and longs. This happens automatically only in assignment contexts and there are two conditions to fulfill.

01. Right hand side value should be a compile time constants
02. Right hand side value should be in the bit range of Left hand side.

EXPLICIT NARROWING CONVERSIONS (CASTING):

Explicit narrowing conversions, also known as casting, are performed when you, as a programmer, explicitly instruct the compiler to convert a value of one data type to another.

07. When we assign a long data type, which is 64 bits in size, to a float data type, which is 32 bits in size, Java automatically performs an implicit conversion (widening or implicit conversion). A long value can hold larger integer with exact precision, whereas a float value is a floating-point number with limited precision (approximately 7 decimal places). Therefore, if the long value is too large or has a high level of precision, some of the least significant bits may be lost when converting to a float.