1.

Statically Type Language

Statically Typed means the data type of a variable is known at the compile time. That means programmer has to specify the data type of a variable at the time of its declaration.

Dynamically typed Language

Dynamically type data means interpreter assign data type the data type to a variable at runtime. There are no any per-defined data types. Types are checking at rum-time.

Java belong to statically and Dynamic type both.

Strongly Typed Language

Types are Checking Strictly. It is necessary to specify the data type of a variable.
Java belongs to the Strongly typed language.

Loosely Typed Language

Data types are not checking Strictly. Loosely typed data types are not strongly bound with data types.

2.
Case Sensitive

case sensitive means characters that are in lower and uppercase treated as distinct. That means distinguish small and Capital letters. Java is a Case sensitive language.

Case-Insensitive

case Insensitive means language is not be able to distinguish between Upper and Lower case versions of a letter.

Case Sensitive-Insensitive

3.

Identity Conversion

A Conversion from a type to that same type is permitted for any type.

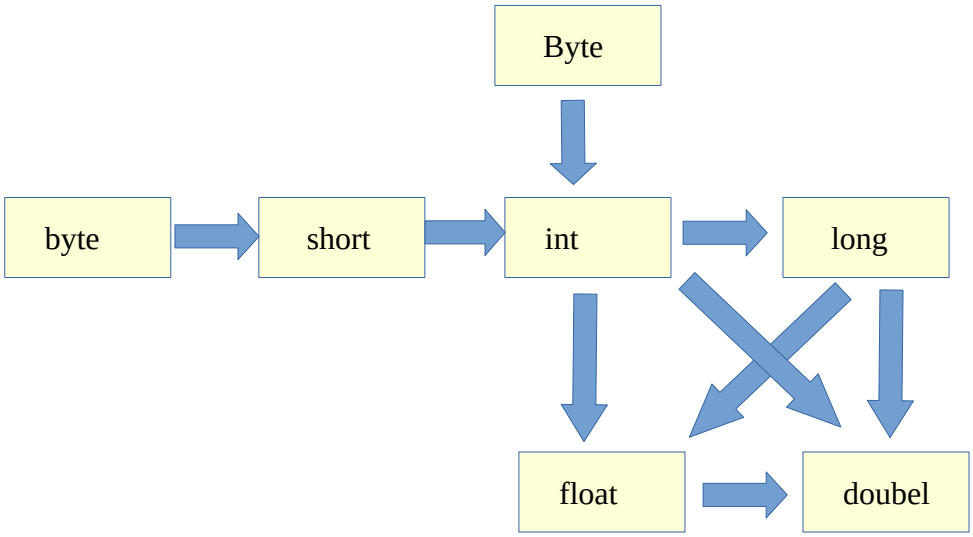
```
int y=10;
int x;

x=y;
x=(int) y;
```

4.

Widening Primitive Conversion

Converting a lower data type into higher data type called widening conversion is done automatically.



EX:-

```
byte myByte=10;
short myShort=myByte;
int myInt=myByte;
long myLong=myByte;
float myFloat=myByte;
double myDouble=myByte;
```

5.

Run-time Constant

Runt time Constant is a value that is computed only at the time when the program is running.run-time constant bit slower than compile-time constant.JVM knows the value of constant in the run-time environment.

Compile-time Constant

These constants knows the values that store,that means compiler know the values inside the constant when compile.

```
final int CONST=10;
int x=10;
```

6.

when converting data types from higher to lower, called narrowing primitive conversions. Implicitly, conversion means a data type converts automatically from a higher data type to a lower data type.

In the casting, the data types are not being converted from higher to lower data types. Those have to be converted explicitly.

Conditions for implicit narrowing conversions

- It should be an assignment Context.
- Constants should be compile-time constants.
- The value should be in the bit range of the data type that is going to be converted.

7.

The method of data storage is different for integers and Floating numbers. Therefore, long data types can be stored in float data types, even 32 bits. Floating numbers can store huge values, but their accuracy is less than that of int-type numbers.

8.

Most programming needs can be fulfilled by using 32-bit integers, which provide a good balance between memory efficiency and the range of values that can be represented. Larger data types, such as long, would unnecessarily consume more memory, while smaller data types, such as byte and short, might result in frequent type conversions.

9.

Due to the size difference, when a larger data type is converted to a smaller data type, some information may be lost. Because the ranges of the mentioned integral data types are clearly defined and it is generally safe to execute without data loss or loss of precision, Java permits this narrowing conversion for those data types.

10.

Both data types in the short-to-char conversion have a 16-bit value. However,the short can only represent the highest positive value of 32767 because the short uses 1bit for the sign bit. Char may represent 65535. The number's positive or negative sign is not stored by the char data type.because the data type char is an unsign Integer. The conversion from short positive value to char is always a

widening conversion.because the value that comes from short is always less than the value of char. Then the value can be stored in 15 bits.