1.Java is Statically and Dynamically Typed and Strongly Typed Language.
Statically Typed Language – Consider data type at compile time.
Dynamically Typed Language – Consider data type at run time.
Strongly Typed Language – Consider data type strictly.
Loosely Typed Language – Consider data type loosely.

2.Case sensitive – ability to identify difference between upper case and lowercase letters.
Ex – Java.C#,C

Case insensitive – Not ability to identify uppercase and lowercase letters and doesn't matter.
Ex- ABAP,Ada,Fortran,SQL

case sensitive-insensitive – uppercase and lowercase doesn't matter.

3.Identity Conversion – A conversion from a type to that same type is permitted for any type. Assigning two instance of same type is identity conversion.

```
Integer i1;
Integer i2 = new Integer(2);
i1 = i2;
i1 = (Integer) i2;

double d1;
double d2=new Double(56);
d1=d2;
d1=(Double)d2;
```

4.Primitive Widening Conversion – Put small value into bigger value range. Does not lose information about the overall magnitude of a numeric value. There is no cast required.

```
byte => short,int,long,float,double
short => int,long,float,double
int => long,float,double
long => float,double
char =>int,long,float,double
float => double

Ex : int myInt=myByte;
     int myInt=myShort;
     float myFloat=myInt;
     float myFloat=myByte
```

5.Run time constant – identify constant at run time
Ex- inal int MY_CONST=10*(int)Math.random();

Compile time constant – identify constant at compile time.
Ex-final int CONST=10;

6.Implicit narrowing conversion(Automatic) – Consider compile time constant and right side value is in left side type value range. It works in automatically.

Explicit narrowing conversion(Casting) – There are use to convert value of wider to narrow. It does not work in automatically.

9.can cause a loss of information, so are generally required to be explicitly cast.