1.
Statically Type Language-language check the type during compile time.
Dynamically Type Language-language check the type during run time.
Stronly Type Language-language that strongly consider the data type.
Loosely Type Language-language that doesn't care much about the data type.

So Java is Strongly typed and Statically typed language.

2.
Case Sensitive->Distinguishes between uppercase and lowercase characters in string comparisons.
example: "IJSE" and "ijse" are considered different.

Case Insensitive->Ignores the difference between uppercase and lowercase characters in comparisons.
example:"IJSE" and "ijse" are considered as same.

Case Sensitive-Insensitive->Provides an option to choose
example: "IJSE" and "ijse" can be considered same or different.

In java ,the defualt behavior for String compertion is case sensitive.

3.
Identity conversion means exchange value between the same data type.
example: boolean flag1 =true;
        boolean flag2 =flag1;

        int int1 = 10;
        int int2 = int1;

4.
Primitive widening conversion in Java automatically converts smaller data type to larger dat type without any data loss..

```
byte myByte = 10;
short myShort = myByte; // myByte value assign to myShort value
```

'/home/gayum/Downloads/image.png'

5.
compile time constant-value that can be determined by the compiler at compile time itself.

```
final String GREETING = "Hello";
final double PI = 3.14159;
final int NUMBER_OF_MONTHS = 12;
```

run time constant-value that is determined and assigned during the execution of the programe at run time

```
final int MAX_VALUE = computeMaxValue();
```

6.
Implicit (Automatic) Narrowing Primitive Conversions
=> automatic convert larger data type to smaller data type
=> may lead to data loss if the value exceed the range of the target data type

Explicit Narrowing Conversions (Casting)
=> manually convert larger data type to smaller data type
=> required explicit casting using perenthaeses and specifying the target data type
=> potential data loss if the value exceed the range of the target data type

conditions for an implicit narrowing primitive conversion
=> conversion is done by complier automatically
=> target data type can represent the entire range of the sourde data type

7.
in float and double the value is represented using scientific notation. So the accuracy of the number reduces.

1.234456 ———> 1.23 x 10^7 but not accurate.
that’s how we can store a long in float or double.

we don’t use float or double when we need accuracy.