01.

Statically typed : Type checking at compile time
Dynamically typed : Type checking at runtime

Strongly typed   : Strongly consider about the type
Loosely typed   : Not strongly consider about the type

Java fall both Statically and Dynamic typed.
Java Strongly consider about the type.
Therefor java is a Statically ,Dynamically typed and Strongly typed language.

02.

Case Sensitive   : A programming language ability to distinguish between upeer and lower case versi0on of letters.

```
                  ex :- int quantity = 50;
                        int Quantity = 20;
```

quantity and Quantity are two different identifiers in case sensitive languages.

Case Insensitive   : A programing language ability to ignore the differance between upper and lower case version of a letter.

```
                  ex :- String name = "Kamal";
                        String Name = "Nimal";
```

name and Name are same identifiers in case insensitive languages.

Java is a case sensitive programming language.This means that java differentiate between uppercase and lowercase letters in identifiers such as variable names ,methods names,class names and keywords.

03.

Identity conversion   : A conversion from a type to that same type is permitted for any type.

```
                        ex :- int x1 = 70;
                        int x2;
                        x1 = x2;//Identity conversion

                        String name ;
                        String familyName = "Perera";
                        name =  familyName ; // identity conversion
```

04.

Primitive Widening Conversion :19 specific conversions on primitive types are called the widening primitive conversions.

-byte to short,int,long,float or double
-short to int,long,float or double
-char to int,long,float or double
-int to long ,float, or double
-long to float or double
-float to double

A widening primitive conversion does not loss information about the overall magnitude of a numeric value.

```
                        ex :- byte age = 30;
                              short myAge = age;
                              int personAge = age;
                              long herAge = age;

                              char a = 65;
                              int letter1 = a;
                              long letter2 = a;

                              float price = 236.23f;
                              double amount = price;

                              float hisAge =age;
                              float hisAge2=myAge;
                              float hisAge3=personAge;
```

05.
A complile time constant is computed at the time the code is compiled, while a run time constant can only be computed while the appilication is running.

```
final int MY_CONST = 10;                    // compile time constant
final int MY_CONST_2 = 10 * (int) Math.random(); // run time constant
```

06.
Implicit narrowing primitive conversion   : Value of a smaller data type is automatically and safely promoted to a value of a larger data type without any loss of information.
Java performs these conversions automatically and no explicit action is required from the programmer.

Explicit narrowing conversions   : Value of a larger data type is explicitly cast into a value of smaller data type.
This operation may results into a loss of information since the destination data type may not have enough capacity to represent the full range of the source data type.

Conditions for an implicit narrowing primitive conversions to occur   :
1.should be a constant
2.The range of values that can be represented by the source data type is within the valid range of values for the destination data type.

07.
When assign a long value to a float,jaav performs an implicit narrowing conversion.
It discards the least significant 32 bits of the long value,and the remaining bits are used to represent the float value

08.Because of the balance between performance,memory usage and practicality.
'int' is the most commonly used data type for integers and it provides a reasonable range for most integer values encountered in typical programming scenarios.
using 'double ' as a default data type helps avoid unnecessary precision loss during calculations.

while 'int' and double are the default data types, java provides support for other data types like long,float,short,byte etc.This allows developers to choose the appropriate data type based on their specific needs,ensuring flexibility and precision when required.
The design decision to use int and double as default dta types in java was influenced by hardware considerations,performance,simplicity and compatibility goals.

09.
Because these data types form a subset of the integer data types that can be safely and efficiently converted without loss of information(smaller sizes,common use cases ,efficiency considerations).

They provide a convenient way to work with smaller integer values without the need for explicit casting in various programming scenarios.

10.widening and narrowing primitive conversion : Conversion combines both widening and narrowing primitive conversions,

byte to char :- First, the byte is converted to and int via widening primitive conversion and then the resulting int is converted to a char by narrowing primitive conversion.

convert short to char can be done directly.(by using widening premitive conversion).
There for conversion from short to char isn't classify as widening and narrowing primitive conversion.