**Programming Fundamentals -
Assignment 2**

1.
Statically typed language - type of programming language where variable types are declared and checked at the compile time rather than runtime. Here we need to explicitly specify the type of variable when we declaring it. Types are checked at the compile time and cannot proceed further with errors. Examples for statically typed languages are java,c,c++

Dynamically typed language – Here variable types are checked and declare at run time rather than compile time. Here we do not need to explicitly specify the type of variable. Variables in dynamically typed languages values different types . More flexible than statically dynamic types. Examples are python,PHP

Strongly typed - Strictly concern about the type of variable. Language not perform automatically type conversions
Loosely typed – Data types are not strictly enforced. Language handles the type conversions.

Java is statically typed and Strongly typed

2.
case sensitivity – Ability to distinguish between upper and lower case versions of letter. Eg – C#,Java, Ruby, XML

Case insensitivity - Ability to ignore the difference between upper and lower case versions of a letter. Eg- Ada, FORTRAN,SQL, Pascal

case Sensitive-Insensitive – Exhibit both case sensitive and insensitive behaviors fo different parts Eg – JavaScript

java is a case sensitive language.

3. A conversion from a type to that same type. This is allowable for any type. This has two important practical implications.
    Expression Permission – Expression always allow to have the desired type right from beginning. No need to use value as it is.
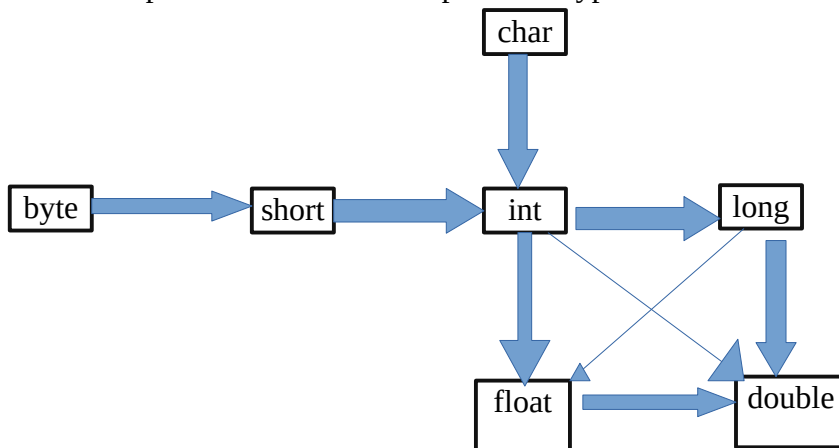
    Redundant Cast operators – Acceptable to include redundant cast operators in a program(In the case of identity conversion)

Example -
    a.   int num =45 ;
            int newNum = (int)num ;
    b.   double myDouble = 3.14 ;
            double newDouble = (double)myDouble ;

4.

19 specific conversions on primitive types are called the widening primitive conversions

```
                        ┌──────┐
                        │ char │
                        └──────┘
                           │
                           ▼
┌──────┐    ┌───────┐   ┌─────┐      ┌──────┐
│ byte │──▶ │ short │──▶│ int │────▶ │ long │
└──────┘    └───────┘   └─────┘      └──────┘
                         │    ╲    ╱    │
                         ▼      ╳       ▼
                      ┌───────┐    ┌────────┐
                      │ float │──▶ │ double │
                      └───────┘    └────────┘
```

Widening primitive conversion does not lose information about the overall magnitude of a numeric value in where

- integral to another integral type
- byte ,short or char to a floating point
- int to double
- float to double in Strictfp expression
   float to double that is not Strictfp may lose precision

Widening primitive conversion may result in loss of precision in following cases.In this case, the resulting floating-point value will be a correctly rounded version of the integer value, using IEEE 754 round-to-nearest mode

- int to float
- long to float
- long to double

Examples

byte myByte = 10;
int myInt = myByte ;

long myLong = 109093;
float myFloat = myLong ;

5.

Compile-time constant - value of the constant is known at the compilation phase of the program. Therefore value does not need to change at the run time.

Example – int myNumber = 2+2 ;

Run-time constant – Value of the constant is generated at the running phase of the program. Program need to run to generate the value.

Example – byte myByte3 = 10 * (int) Math.random();   // Run time constant

6.

Automatic NPC - Value of the wider data type assigned to narrower data type implicitly. –This occurs under two conditions.

- Value to be assigned(right hand side) should be a compile time constant.
- Value (constatnt) should be fall within the bit range of the type that going to assign.

Example – int myInt = 100 ;
                byte myByte = myInt ;
Castings - Casting is necessary when compiler implicitly cannot convert wider data type to a narrower data type.

7.
Long is a 64 bit two ' s complement integer. It can consists of maximum value of 2^{63-1} and maximum of 2^{63}. Float data type is a single precision 32 bit IEEE 754 floating point. It can hold not only positive and negative numbers that consist of a sign and magnitude, but also positive and negative zeros, positive and negative infinities, and special Not-a-Number values. Float cannot use for precise values. But it can hold larger values even with 32 bits.

8. In present most of the computers contain 64 bit architecture in CPU. Very few with 32 bit hardware architecture also may present. These CPU are optimized for perform arithmetic operation of fixed size data types. Therefore performance is high.

9.Prevent data loss. Precision issues come across with other types.

10. Combines both widening and narrowing primitive conversions.
Example
Byte to char – char is unsigned integer and byte is signed integer. First byte is converted to an int via widening primitive conversion and then resulting int converted to char using npc.

Type char represents Unicode characters, Which are unsigned and include broader range than short. When conversion of short to chat it maps the numeric values to corresponding unicode character rather than just discarding bits.