Core Java :

1. Fundament of the java
2. With only core java we cannot develop a app
3. But same core java is use in advance part so the learning core java is very important

Features of java :

1. Java supports OOPs
2. Dynamic programming
   1. We can create the object in run time
   2. We can remove the object in run time (garbage collection)
   3. User need to create the object
   4. Garbage collection is auto
   5. When the object is de-referred the object will go for garbage collection
3. Multi threading : multiple users can access the class at the same time which is also called concurrent programming
4. Robust: different type of application can be developed using java like
   1. J2ee,
   2. Spring boot microservice
   3. Socket programming
   4. Messaging
5. It is a platform independent
   1. Once we write code in one OS we run on any OS without any changes
   2. This two level one level is **compilation and interpretation**
   3. User write x.java 🡪 compiled to x.class 🡪interpreted by jvm 🡪 machine code 🡪 executed by os
   4. Java 🡪 bytecode🡪 machine 🡪 executed
   5. Since the machine code is not saved to execute the java each time we need jvm to interpret
   6. The jvm is different for each os (jvm is platform dependent )
   7. Write once execute any where

The different version of java are

1. Java 1.6
2. Java 1.7 more on unanimous (class with out name )
3. Java 1.8 is base version of all higher version
   1. Lambda expression
   2. Functional programming
   3. Stream
   4. Multi thread
   5. File stream
4. Java 11,13,17 19, 20
5. LTS java 1.8 ,11, 13,17

Executing java :

1. We need java kid (JDK) this has got compiler and interpreter
2. JRE interpreter
3. Javac and filename with .java will compile the code
4. Java and filename without .java
5. Before we starting many of the tool we should first install the jdk and environment variable
6. We can also use Maven build tool which compile and build the java code it will add the dependency automatically
   1. POM project object module
   2. It will compile
   3. It will test
   4. It will do code coverage
   5. Modularity
7. Equivalent for maven is gradel ANT

Java rules (Note if we do not follow the following rules it is compilation error )

1. It is case sensitive we need to write the code in lower case
2. ; as eol
3. In one file only one public class is allowed
4. Suppose if we need more public class we need create in separate files
5. The public class name should the file name
6. The java can be executed with main or with out main method / function
7. The signature of the main method is
   1. Public static void mian(String[]arg)
   2. This is the reason the java is not a complete object oriented
8. Here String is also immutable class (when we do the changed the new object is create)
9. Data type is
   1. Primitive (if any language support primitive they are not complete oops)
   2. Wrapper class

|  |  |
| --- | --- |
| Primitive | Wrapper class |
| I) int(4) , float(4) , char(2), long(8), double (8) -  -boolean | Integer ,Float ,Charater, Long, Double Boolean |
| Good for complex computation | Storage and collection |
| External type conversion | Has methods to convert from one data type to another |
| - float x  Int y = (int) x | Float x  Interger y  y.valueof(x) |

Note String is not a data type it is class

|  |  |
| --- | --- |
| Class | Object |
| Is blue print of the entity which contain data and method but not ready to use | A memory allocated to the class it is ready to use |
| First we need to create a class | Then the object  For the same class we can create multiple objects |