16/12/19

UNIT-1 INTRODUTION TO OOP

* Java is an Object Oriented Programming language (OOPs).

C → C++ → Java.

C → structured prog lang

C++ → Object oriented Prog lang.

Java

→ C ⊆ C++ ⊆ Java.

C → functions

C++ → classes & Objects

Java → classes & Objects

Java is platform independent.

Difference b/w C++ and Java:

C++ → without class, with class → partially object oriented prog lang.

Java → with class. → OOPL

* main() is compulsory.

* In C++, all the classes are written first and at last main() is written. main() is written outside the class.

In Java, main() is included at one of the class.

C++

```
class A
 { _
 }
class B
 { __
 }
void main()
 {
 }
```

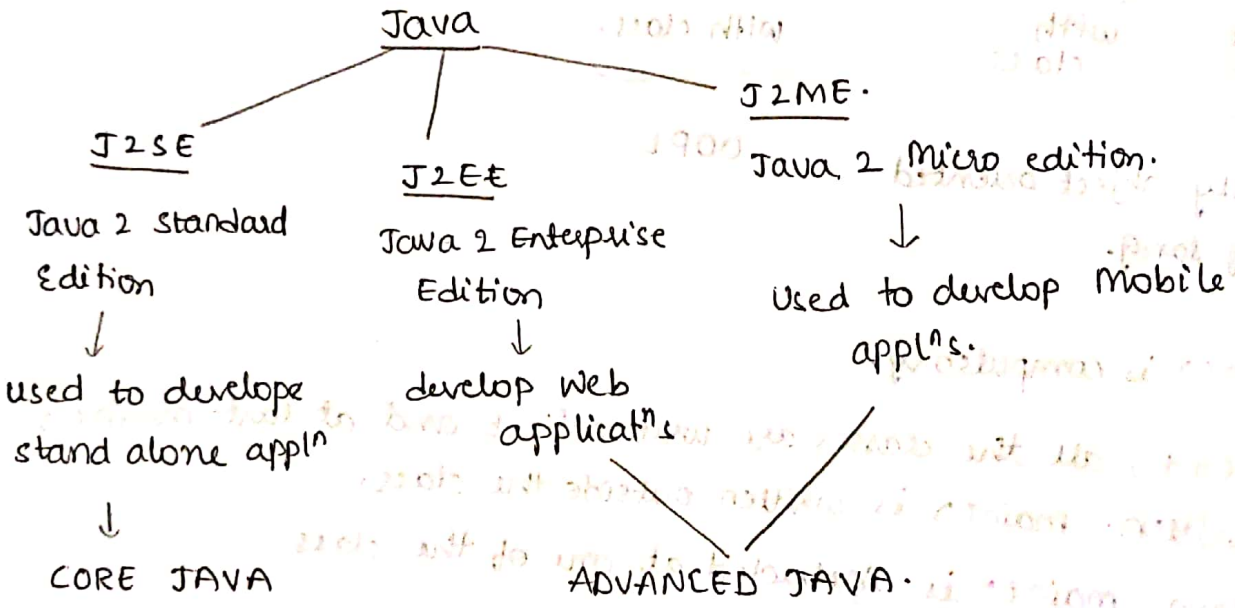
Java

```
class A
{  void main()
    {
    }
}
class B
 {
 }
```

C++ Vs Java:

- We can write C++ programs with or without classes.
  Java programs are written only with classes.
- In C++, main() is written outside the class.
  Java, main() is written inside any one of the class.

* Java was developed by James Gosling in the year 1995 then, sunmicrosystems.
  Initial version of Java is 1.0.
  current version -



Stand alone applns:
Client and server are same.
No servers are seperately available.

C prog is also stand alone appln.

Web appln:
One server and many clients.
Eg: Google.

J2EE and J2ME depend on core JAVA.

J2EE and J2ME are independent.

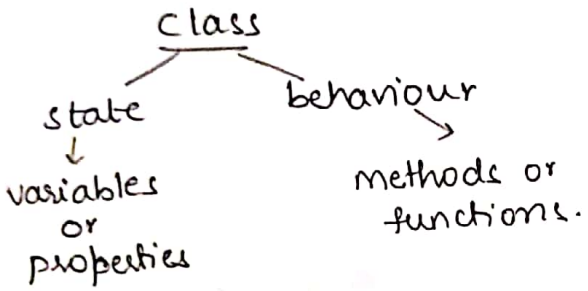
* Class and Object:

Class is a template | blue print of an object.

Object is an instance of the class.

syntax

```
class classname
  {
    state
    behaviour
  }
```



class may contain
- state
- only behaviour
- Both state and behaviour.

→ C language cannot be used for developing real time applications. The main advantage of Java over C is Java is used for Real time applications.

Class Eg for students:

```
class student
{
  int rno;
  string name;
  float marks;
  void displaydetails()
  {
  }
}
```

state

behaviour

string - predefined class
↓
same as character array in C.

For employees:

```
class employee
{
  int empid;
  string name;
  int number;
  void display()
  {
  }
}
```

For Box:

```
class box
{
  int length, breadth, height;
  void setvalues()
  {
  }
  void displaydetails()
  {
  }
}
```

state

beh

Run time errors are Exceptions in Java.

Simple Java program:

```
class A
{
  public static void main(string args[])
  {
    System.out.println("Hello to Java");
  }
}
```

only behavioural part.

- save the file [classname.java] A.java

compile - javac is the compiler. [javac classname.java]

Execute - [java classname]

java A.

* Mains can be written in any number of classes.

Prog:

```
class A
{
 public static void main (string args[])
  {
   System.out.println ("Hello to Java");
  }
}

class B
{
 public static void main (string args[])
  {
   System.out.println ("Hello to Java B");
  }
}
```

- Here, we can save the file with any classname.
The execution takes place for the classname given.

O/P:

saving as A.java.

Hello to Java.

saving as B.java

Hello to JavaB.

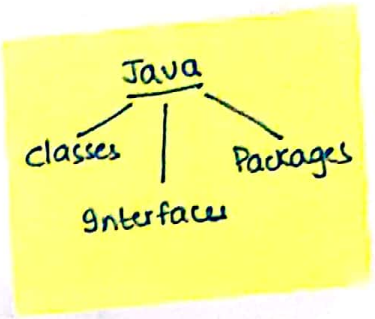
18/12/19

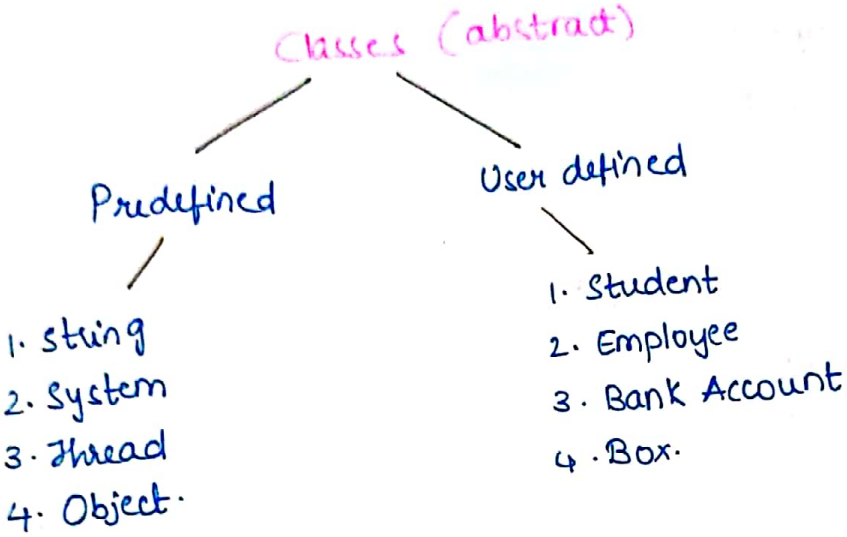
* Containers in Java:

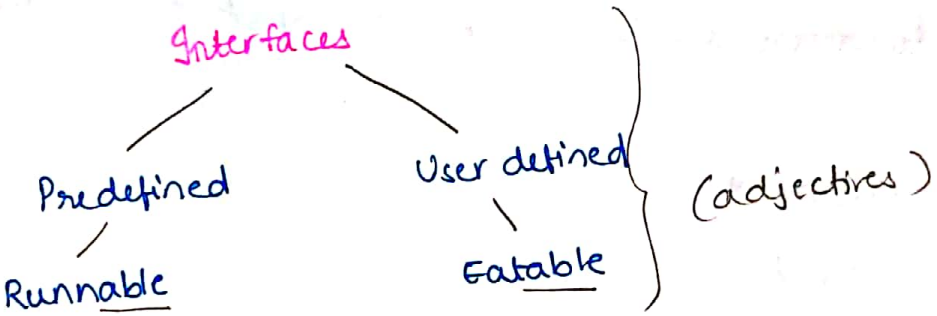
There are 3 containers in Java.

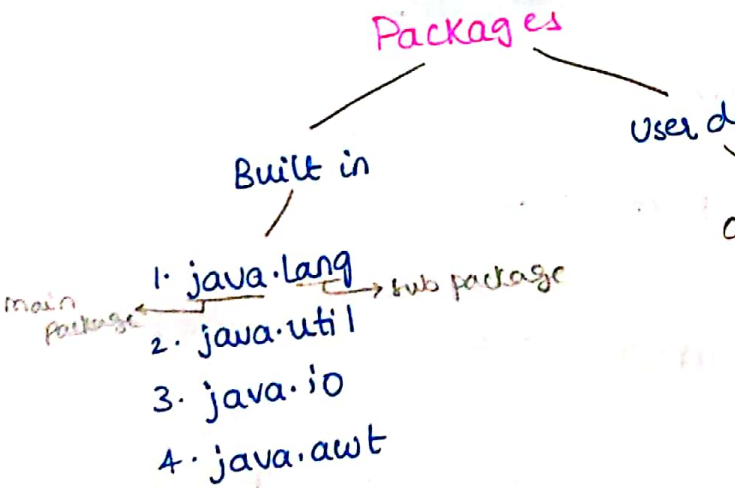
1. Classes
2. Abstract classes
3. Interfaces.

```
string name = "ramu"
System.out.println("Hello");
```

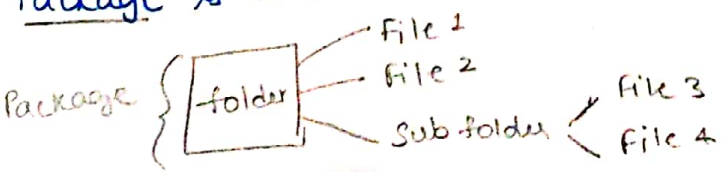








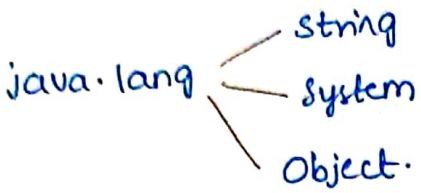
Package is a folder that contains classes and interfaces.

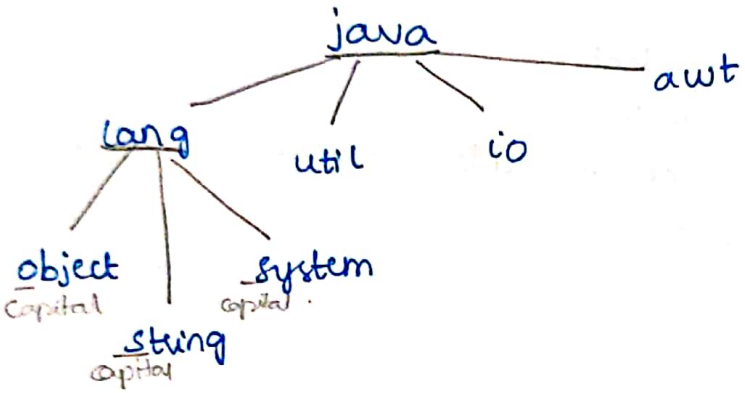


Predefined Packages consists of Pre-defined classes and Pre-defined interfaces and Pre-defined Sub Packages.

Eg: java.lang.

classes of lang:





* java.lang is the default package in Java.

-If we want to access all the classes or import the classes of lang.

```
import java.lang.*;
```

For string | object | system

```
import java.lang.Object;
```

(Object: capital)

For string and system

```
import java.lang.String,System;
```

* Every class in Java extends object class

extends object → default

* First statement in any Java program is

```
import java.lang.*;
```

But since, it is a default package, it is not necessary to write the statement.

But for any other packages, we need to mention it explicitly.

Object class:

```
public class Object
    {
      public string to string ()
      { =
        }
      public int hashCode ()
        {_
          _
          }
    }
```

capital

Java does not support multiple inheritance
↳ extending
getting the methods from one class to other.

- Multiple inheritance:
  C++ supports
  java does not support.
- Multilevel inheritance:
  Java supports.

* Naming Conventions:

→ Classes: (Rules for classname):

- If there is only one word then the first letter of the word must be capital.

  Eg: Student,

- If it contains more than one word the first letter of the first word must be capital and also the first word of the subsequent words must also be capital.

  Eg: class JavaDemo { }
  class StudentDetails { }.

→ Method:

- First word first letter must be small (lower case)

  Eg: void display { }

- More than one word, first letter of the first word must be lower case and the subsequent words first letter is capital.

  Eg: void displayDetails { }

→ Interfaces:

Interfaces rules are same as class.

Eg: interface Eatable
interface Runnable.

→ Packages :

All the letters in packages are small (lower cases)

Eg :

```
package pack1;
```

After compilation in Java, class file is created.
When we open that, we get the byte code.

- Byte code is used to copy it and execute it at any other system.
  Hence, it is platform independent.

→ JVM (Java Virtual Machine)
  It is used to refer to the class file without going to java file.
  It goes to the class file.

- In a Java program, we can write any no. of classes.

Program:

```
class A
{
public static void main(String args[])
 {
  System.out.println("Hello to Java");
 }
}

class B
 { int a;
   void m1()
    {
     System.out.println("m1()");
    }
 }
```

Saving the file as A.java.
- compiling A.java
  ↓
  A.class
  B.class
- Execution occurs in A. since, it searches for main method.
  O/P: Hello to Java

Saving the file as B.java.
- compiling B.java
  ↓
  A.class
  B.class
- Execution occurs, but since no main method is used.
  Runtime error occurs → Exceptions

- If there are two or more classes, save it with either A.java or B.java.
- While compilation, all the classes .class files are created.
- Execution starts from the main method.
  JVM searches for main.

→ If it is saved for with any other name, runtime error occurs.