JAVA

Java History:

Java intro - James Gosling -1991 to 1994.

developed by Sun Microsystem - 1995.

Naming History :

First designed in embedded systems[hardware +software] in the name of oak in 1991 and renamed as Java in 1995.

1.Oak – tree name

2.Java - coffee

1. What is Java?

-> Popular programming language

-> Writing, compiling and debugging is easy

-> It can be reused

2. Java features (or) importance of java (or) why we go for java?

-> Platform independent (JAVA compiler converts source code into byte code, can be executed in any platform.

-> Open source (free of cost)

-> Secured (virtual firewall between application and the computer)

-> Multi-threaded (more task can be completed at a time and Multitasking - different task can be completed at a time)

-> Portable (write once run anywhere) WORA

3. Java architecture:

JDK: (JAVA DEVELOPEMENT KIT)

-> To develop JAVA code, JDK has to be install

-> It has development kit

-> JDK version 1.0 to 1.16, 1.17 is in progress

-> Stand version is 1.8

JRE: (JAVA RUNTIME ENVIRONMENT)

- it helps to run the program

-> it contains predefined library files (.class files)

JVM: (JAVA VIRTUAL MACHINE)

-> it is used for memory allocation

-> it is not physically exists.

JDK= Development kit + JRE

JRE= Predefined files+ JVM

4. Coding standards:

There are two types of coding standards:

- Pascal
- Camel

PASCAL: Every words First Letter must be in Capital

Eg: Employee_Details- (used in Project name and class name)

CAMEL: Except first word remaining words First letter must be Capital.

Eg: employeeDetails- (used in Object ,method and variable).

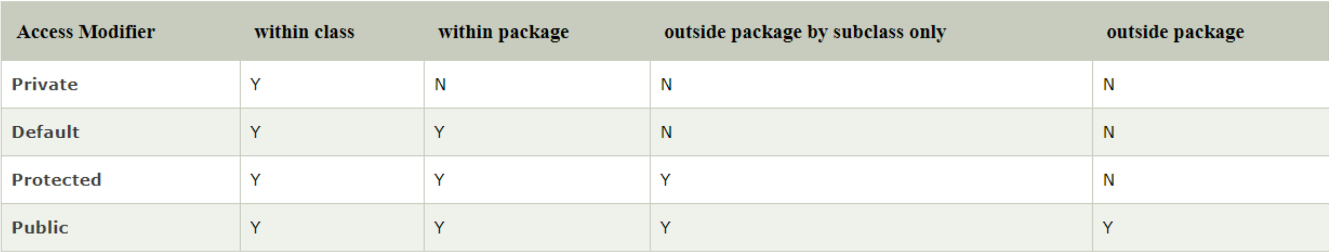
Eclipse Project Creation:

FILE-New(Ctrl+N)→java→JavaProject→ProjectName(Pascal)→Next→Finish

Package Creation:

Src(R.C)→New→Package→Name→Finish

Class Creation:



Access Modifier	within class	within package	outside package by subclass only	outside package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y

Package(R.C)→New→Class

5. Access Specifier:

- **Public:** Global level access (inside and outside the package)
- **Private:** Class level access (inside the class)
- **Protected:** same like public but use **“extends”** keyword
- **Default:** Package level access.

6. Garbage collection:

- Un-referenced object are deleted automatically.
- Un-wanted or un-used memory are deleted automatically.
- It is possible only in java not in c, c++ etc. eg:finalize();

7. Datatypes:

To specify the type of data…

Syntax:

Datatype variableName = Value;

Two types:

- Primitive : - 1. A variable can store only one value
  2. Predefined datatype.
  3. There is no additional methods.
  4. Store in memory reference.
  5.Types : 8 types….byte, short, int, long, float, double, char, boolean.

- Non Primitive :- 1. A variable can store group of values.
  2. Based on class
  3. It is reference variable (or) object reference

4. Store based on reference.
5.Types : 2 types : Array and String.

DATA TYPE	MEMORY SIZE	DEFAULT VALUE	WRAPPER CLASS
byte	1 (8 bits)	0	Byte
short	2 (16 bits)	0	Short
int	4 (32 bits)	0	Integer
long	8 (64 bits)	0	Long
float	4 (32 bits)	0.0f	Float
double	8 (64 bits)	0	Double
char	2	-	Character
boolean	True/false	false	Boolean

8. Wrapper Class:

- Convert datatype into class object
- It is used in collection.

9. Range of datatype:

-2^(n-1) to 2^(n-1)-1

Eg:

1. byte : 8 bits= 1byte

   n=8

   -2^(8-1) to 2^(8-1)-1

   -128 to 127

2. int : 32 bit= 4 byte

   n=32

   -2^(32-1) to 2^(32-1)-1

10. OOPS:- Object Oriented Programming Structure (or) System.

Method of implementation in our program are organized in the form of method, object and class.

Class: Combination of object and method.

Method: ->Set of actions to be performed.

->Collection of statements within curly braces. {}

->Code complexity is reduced...

Object: -> Instance of the class.

->helps to call the method

-> it allocates memory

->new keyword for creation/allocating memory.

->object is created in main method only.

Syntax :

```
    ClassName refName/objName=new ClassName();
```

Eg:

```
Public class StudentDetails{
     Public void stdName(){
       System.out.println("Arun");
}
     Public void stdId(){
      System.out.println("123");
}
Public static void main(string[] args){
     StudentDetails sd= new StudentDetails();
     sd.stdName();
     sd.stdId();
}
```

11.Oops Concepts :

Encapsulation

Inheritance

Polymorphism

Abstraction

12.Encapsulation :

The process of Binding Code And Data together A Single entity(information).

Creating a structure of folders.

Code = class+varibles+objects+methods...

Data = given data...

13. Inheritance:

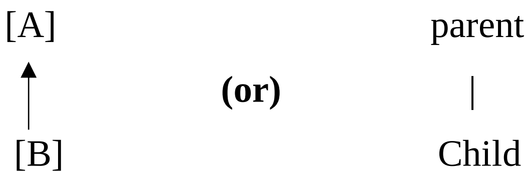
- Relationship between parent and child class.
- One class access the property of another class using **“extends”** keywords.

Types of Inheritance:

- Single
- Multiple
- Multilevel
- Hierarchical
- Hybrid

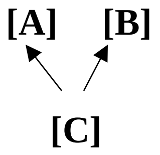
Single Inheritance:

- Child class (B) access the property of parent class (A).

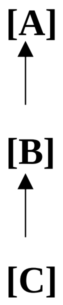


Multiple Inheritance:

- It has more than one parent class is accessed by one child class.
- So there will be priority issue to access the parent class.
- It is not possible in JAVA.
- can overcome this disadvantage by using **“Interface”.**



Multilevel Inheritance:



- one parent class will be accessed by one child class that will be accessed by another child class.
- Ex :

Parent - singing

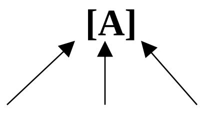
|

child - dancing, singing

|

Grand child - sports, dancing, singing

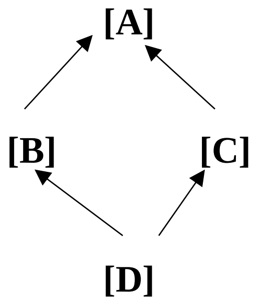
Hierarchical Inheritance:



[B] [C] [D]

- One parent class will be accessed by more than one child class...

Hybrid Inheritance:



- Combination of Single inheritance and Multiple Inheritance.
- Not possible in java.