LECTURE - 1

Introduction to Object Oriented Programming:-

Introduction of java:- (Developed By James Gosling in 1995 at Sun Microsystem)

It is pure object oriented programming languge which must be support following Four properties.

The history of Java is very interesting. Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time. The history of java starts from Green Team. Java team members (also known as Green Team), initiated this project to develop a language for digital devices such as set-top boxes, televisions etc. But, it was suited for internet programming. Later, Java technology was incorporated by Netscape.

The principles for creating Java programming were "Simple, Robust, Portable, Platform-independent, Secured, High Performance, Multithreaded, Architecture Neutral, Object-Oriented, Interpreted and Dynamic"

Currently, Java is used in internet programming, mobile devices, games, e-business solutions etc. There are given the major points that describes the history of java.

- Polymorphism(P)
- Inheritence(I)
- Encapsulation(E)
- Abstraction(A)

Polymorphism:-

A single function may perform various types of tasks depending upon arguments.

Example:-

draw (line)

draw (Circle)

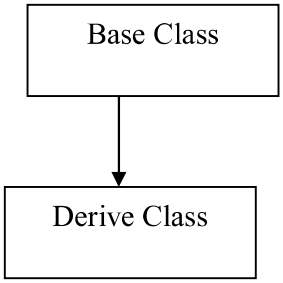
draw (Rectangle)

draw (Oval)

Inheritence:-

It means reusability.That is property of base class is going to access by its derive class.

Example:-



Encapsulation:-

It is an abstraction process of binding and hiding properties.

Abstraction:- It is representation of essential features of a system without involving the complexity of system.

Concepts of Class & Objects:-

Class:-

It is combination of **data member (Attributes)** and **member functions (Behaviors)**.It has no physical existence.

Example:-

Class person.

Class Animal.
Class Vehicle.
Class Vegatable.
Class Students.
Class Teachers.
Class Leader.
Class Country.
Class Temple.
Class Furniture.
Etc.

Object:-

It is an instance of class is called object.It has a physical existence.

Example:-

Amrita is an object of class person.
Sujita is an object of class BCA II Year.

Format of class Declaration:-

```
class <class_Name>
{
private :
        Data_members;
        Member_functions;
public :
        Data_members;
        Member_functions;
protected :
        Data_members;
        Member_functions;
};
```

Where :-

private, public and protected are called **visibility mode**.(VM)
Private and protected cannot access directly by object of that class in main function.
Public member function and data member directly access by object of that class in main function.

Example of object oriented language:-

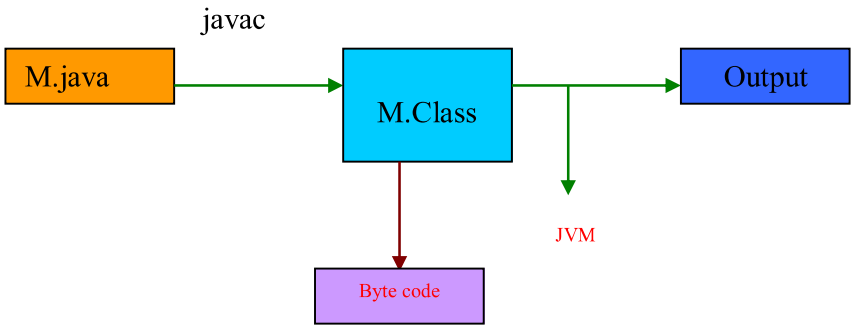
C++, JAVA, Smalltalk, Beta, C#, Python etc.

Feature of OOPs:-

- Reusability (Inheritence).
- Polymorphism.
- Encapsulation.
- Generacity.
- Data abstraction.
- Distributed programming perform.
- Robust.
- High security.
- Portable.
- High performance.
- Multithreaded programming(Threads means process).
- Scripting programming(java script).
- Architecture neutral(Only in Java).

Java Compiler:-

Java compiler is a s/w that converts the java's program into the machine language. and converts them into byte code.It can perform by javac (java compiler).Java compiler can be driven by **java virtual machine (jvm)**



Concept of Message:-

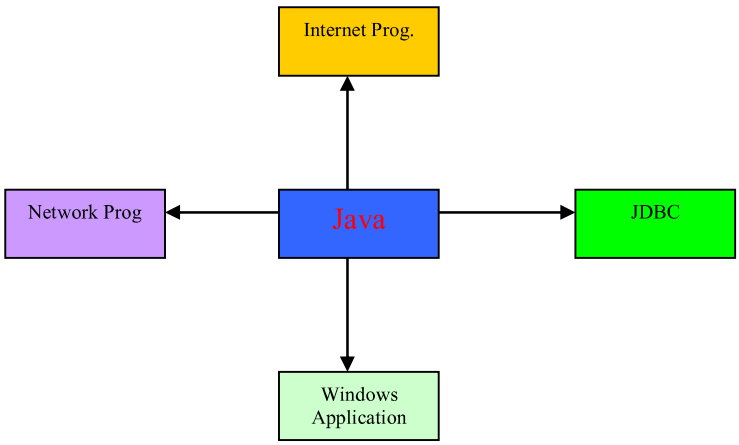
Note: - S/w objects interact & communicate with each other using message.

Example:- Object A want object B to perform one of B's methods,
Object A sends a message to object B.

There are three types of library in java.

- Language Class.
- Utilities Class.
- I/O class.

Use of Java



JDK: - (Java Development Kits)

It provides environment for java programmings.Which consist of following files.

- Appletviewer (It is used for displaying effect of java program on browser).
- javac(java compiler).
- java.
- javap(Java Disassembler).
- javah(java header files).
- javadoc(For creating HTML document).
- jdb (Java Debugger).

Example of Simple Java program:-

Example1

```
import java.io.*;
class pr1
 {
  public static void main(String args[])                    //main function of java
   {
   System.out.println("Welcome in strong OOPs language Java");
    }
 }
```

Example2

```
import java.io.*;
class pr2
{
 public static void main(String args[])                    //main function of java
 {
    int a=15,b=16, c1,c2;
    c1=a+b;
    c2=a*b;
    System.out.println("Sum="+c1+"\n"+"Product ="+c2);     //For display on screen
   }
}
```

Java Tokens:- *(Most Important)*

Java class is a set of declaration statements and methods containing executable statements. The smallest unit in java program is called tokens. There are following five types of java tokens.

- Reserved Keywords
- Identifiers
- Literals
- Operators
- Separators

Reserved Keywords/Already built in java library we can not use as variable.

if for break byte case catch try extends native new class continue else int import generic goto return double char static public switch void while throw super package etc.

Identifiers:

It is user defined token. They are used for naming classes, methods, variables, objects, labels, packages and interface in a program.

Rules for identifiers:-

1:-They can have alphabets, digits, underscore and dollar sign characters.
2:- They must begin with alphabet.
3:- Uppercase and lowercase letters are distinct.
4:-They can be any of length.

Example:-

```
int rol_num,Hindi_marks,eng_marks,maths_marks;
char stu_name[40];
float principal_amt;
int stu_age;
```

Literals:-

It is sequence of characters (digits, letters and other characters).
There are five types of literals.

- Integer literals
- Floating point literals
- Character literals
- String Literals.
- Boolean Literals.

Example of Integer literals:-

```
int a=5, b=8, c=12;
```

Example of floating literals:-

```
float a=5.4, b=8.5, c=12.89;
```

Example of character literals:-

```
char a='5.4',m='n',x='7';
```

Example of string literals:-

```
char a[10]="VARANASI";
```

Example of booleans literals:-

boolean literate=1,ilitterate=0;

Operators:-

- Arithmetic operator (+,-,*,/,%)
- Relational Operator(>,<,>=,<=,!=,= =)
- Logical Operator(&&,||,!)
  - && and
  - || or
  - ! Not
- Assign operators (=).
- Bitwise operator.
  - & Bitwise and
  - | Bitwise or
  - ^ Bitwise exclusive or
  - ~ One's Compliment
  - << Shift left
  - >> Shift Right
  - >>> Shift right with zero fill
- Ternary Operator/Conditional Operator (?,:)
  - expr1?expr2:expr3
    expr2 execute when expr1 will be true.
    expr3 execute when expr1 will be false.
- Special Operator
- dot operator(.)
- Increment and decrement operator.
  - ++a preincrement value of a incremented by 1 first then assign in variable.
  - a++ Post increment first assign value in variable then incremented by 1.
  - --a predecrement value of a decremented by 1 first then assign in variable.
  - a-- post decrement first assign value in variable then decremented by 1.

Separators:-

It is used for separating variables, define function, terminating statement, define array size etc.There are many types of separators:-

( ) parenthesis
{} braces
[ ] brackets
; Semicolons
, comma
. Period

How to write program in java

Example:-

```
import java.io.*;
class pr3
{
 public static void main(String args[])
 {
    int a=2,b=3;
    System.out.println("Value of a="+a+"\t"+"Value of b="+b);
    a=++a + ++b;
    System.out.println("Value of a="+a+"\t"+"Value of b="+b);
    a=--a + b--;
    System.out.println("Value of a="+a+"\t"+"Value of b="+b);
   }
}
```

Example:

```
import java.io.*;
class pr2
{
 public static void main(String args[])
 {
    int a=2,b=3;
    System.out.println("Value of a="+a+"\t"+"Value of b="+b);
    a=++a + ++b;
    System.out.println("Value of a="+a+"\t"+"Value of b="+b);

   }
}
```

Example:-2(Bitwise operator)

```
import java.io.*;
class pr3
{
 public static void main (String args[])
 {
    int a=10,b=9,t1,t2,t3,t4,t5,t6,t7;
    t1=a&b;
    t2=a|b;
    t3=a^b;
    t4=~a;
    t5=a<<2;
    t6=b>>2;
    t7=a>>>2;
    System.out.println("Bitwise and="+t1);
    System.out.println("Bitwise OR="+t2);
    System.out.println("Bitwise Exclusive OR="+t3);
    System.out.println("Bitwise Compliment="+t4);
    System.out.println("Bitwise Left shift="+t5);
    System.out.println("Bitwise Right Shift="+t6);
    System.out.println("Bitwise Shift right With Zero="+t7);
   }
}
```

Data Types:- **(Most Important)**

Integer Types

Type	Size	Minimum value	Maximum Value
byte	1 Byte	-128	127
short	2 Byte	-32768	32767
int	4 Byte	-2147483648	2147483647
long	8 Byte	-9223372036854775808	9223372036854775807

Floating Data/Real Data Types

Type	Size	Minimum value	Maximum Value
float	4 Byte	3.4e -038	3.4e +038
double	8 Byte	1.73e-308	1.73e+308

Character Type:-

Type char size 1byte

Scape sequence:-

'\n' new line
'\t' tab
'\f' form feed
'\r' carriage return
'\' ' Single quote
'\" ' double quote
'\\' back slash

LECTURE - 2

How to write program in java:-

```
import java.io.*;
class prg1
{
 public static void main(String args[])
 {
```

```
    int a=10,b=7,c1,c2;
    c1=a+b;
    c2=a*b;
    System.out.println ("Sum of two numbers="+c1);
    System.out.println ("Product of two numbers="+c2);
   }
}
```

How to input values through keyboard:-

First Method:-

Format of try-catch

```
{
        …
        …
    try
    {
        …
        …
        …
      }
      catch (Exception e){}
  }
```

Second Method:- Example1

```
import java.util.Scanner;
class prg10
{
 public static void main(String args[])
{
   int a,b,c;
   Scanner sc=new Scanner(System.in);
   System.out.println("Enter first number=");
   a=sc.nextInt();
   System.out.println("Enter second number=");
   b=sc.nextInt();
   c=a+b;
   System.out.println("Sum of two numbers="+c);
 }
}
```

Second Method:- Example2

```
import java.util.Scanner;
class prg11
{
 public static void main(String args[])
{
 int rollno;
 String name,cname;
 double fee;
   Scanner sc=new Scanner(System.in);
   System.out.println("Enter your rollno");
    rollno=sc.nextInt();
   System.out.println("Enter your name");
    name=sc.next();
    System.out.println("Enter Course name");
    cname=sc.next();
   System.out.println("Enter your fee");
    fee=sc.nextDouble();
```

```
  System.out.println("Rollno of Student:"+rollno+"\n"+" Name of Student:"+name+"\n"+"Name of
Course:"+cname+"\n"+"Course fee:"+fee);
 }
}
```

Example:-

```
        import java.io.*;
class pr4
{
 public static void main(String args[])
 {
    DataInputStream in=new DataInputStream(System.in);
    int a=0,b=0,c1=0,c2=0;
    try
    {
       System.out.println ("Enter value of a=");
       a=Integer.parseInt(in.readLine());
       System.out.println("Enter value of b=");
       b=Integer.parseInt(in.readLine());
       c1=a+b;
       c2=a*b;
      }
      catch(Exception ramayan){}
      System.out.println("Sum="+c1);
      System.out.println("Product="+c2);
   }
}
```

Example:- Division of Any Two Numbers

```
        import java.io.*;
class pr5
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
    float a=0, b=0,c1=0;
    try
    {
       System.out.println("Enter value of a=");
       a=Float.valueOf(in.readLine()).floatValue();
       System.out.println("Enter value of b=");
       b=Float.valueOf(in.readLine()).floatValue();
       c1=a/b;
      }
      catch(Exception Jitendra){}
      System.out.println("Divide value="+c1);
   }
}
```

Example :-Area of Circle

```
import java.io.*;
class pr6
{
 public static void main(String args[])
 {

    BufferedReader in=new BufferedReader(new InputStreamReader(System.in));
    double r=0,a=0;
    try
    {
       System.out.println("Enter Radious Of Circle=");
       r=Float.valueOf(in.readLine()).floatValue();
       a=3.14*r*r;
      }
      catch(Exception Jitendra){}
      System.out.println("Area of Circle="+a);
   }
}
```

Example :-Area of Triangle

```
import java.io.*;
class pr7
{
 public static void main(String args[])
 {

    BufferedReader in=new BufferedReader(new InputStreamReader(System.in));
    double a=0,b=0,c=0,s=0,area=0;
    try
    {
       System.out.println("Enter First side of triangle=");
       a=Float.valueOf(in.readLine()).floatValue();
       System.out.println("Enter Second side of triangle=");
       b=Float.valueOf(in.readLine()).floatValue();
       System.out.println("Enter Third side of triangle=");
       c=Float.valueOf(in.readLine()).floatValue();
       s=(a+b+c)/2;
       area=Math.sqrt(s*(s-a)*(s-b)*(s-c));
      }
      catch(Exception Hariom){}
      System.out.println("Area of Triangle="+area);
   }
}
```

Example:-Conversion of degree centigrade temperature into foreignheight

c=5*(f-32)/9

f= (9*c)/5+32

```
import java.io.*;
class pr8
{
 public static void main(String args[])
 {

    BufferedReader in=new BufferedReader(new InputStreamReader(System.in));
    double c=0,f=0;
    try
```

```
    {
      System.out.println("Enter Temperature in Centigrate=");
      c=Float.valueOf(in.readLine()).floatValue();
       f=(9*c)/5.0+32;
     }
     catch(Exception Hariom){}
     System.out.println("Temperature in Foreign height="+f);
   }
}
```

LECTURE - 3

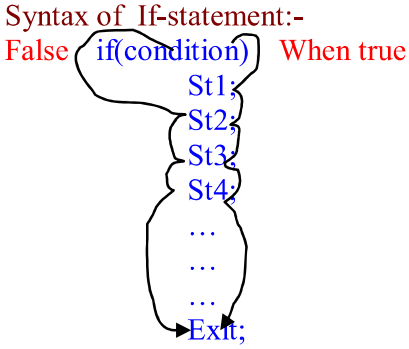
Control Statement:- Important

'Java' language provides facilities for controlling the order of execution of the statements, which is referred to as flow control statements/control statements.

There are following three categories of flow control statements

1) Decision Control Statement
   a. If statement
   b. if-else statement
   c. nested if-else statement
   d. else-if construct statement/Ladder Statement
   e. switch case statement
2) Looping Control/Iterative Control/Repetitive Control Statement
   a. while loop.
   b. do while loop.
   c. for loop.
3) Jumping Control Statement
   a. break statement.
   b. continue statement.

Syntax of If-statement:-



Example :-Check Year is leap year or not:-

```
import java.io.*;
class pr9
{
 public static void main(String args[])
  {

    DataInputStream in=new DataInputStream(System.in);
    int year=0,y=0;
    try
    {
      System.out.println("Enter any Year=");
      year=Integer.parseInt(in.readLine());
      y=year%4;
     }
     catch(Exception Jitendra){}
     if(y= =0)
```

```
        System.out.println("Leap Year="+year);
         if(y!=0)
        System.out.println("Not Leap Year="+year);
    }
}
```

Example:-Quadratic Equation:-

```
import java.io.*;
class pr10
{
 public static void main(String args[])
 {

    BufferedReader in=new BufferedReader(new InputStreamReader(System.in));
    double a=0,b=0,c=0,x1=0,x2=0,d=0;
    try
    {
          System.out.println("Enter Coefficient of X^2=");
          a=Float.valueOf(in.readLine()).floatValue();
          System.out.println("Enter Coefficient of X=");
          b=Float.valueOf(in.readLine()).floatValue();
          System.out.println("Enter Constant=");
          c=Float.valueOf(in.readLine()).floatValue();
          d=b*b-4*a*c;
          if(d>=0)
          {
           System.out.println("Roots are real");
            x1=(-b+Math.sqrt(d))/(2*a);
            x2=(-b-Math.sqrt(d))/(2*a);
           System.out.println("First Root="+x1+"\n"+"Second Root="+x2);
          }
          if(d<0)
          {
           System.out.println("Roots are Imaginary");
          }
       }
       catch(Exception Hariom){}
    }
}
```

Syntax of If-else-statement:-

```
          if<condition>
                    Statement_blocks_True;
          else
                    Statement_blocks_false;
                    Statement_blocks_Exit;
```

Example:-

Example:-Odd/Even Number Checking:-

```
import java.io.*;
class pr11
{
 public static void main(String args[])
 {
```

```
    DataInputStream in=new DataInputStream(System.in);
    int n=0,m=0;
    try
    {
       System.out.println("Enter any number=");
       n=Integer.parseInt(in.readLine());
       m=n%2;
      }
      catch(Exception Jitendra){}
      if(m= =0)
      System.out.println("Even Number="+n);
         else
      System.out.println("Odd Number="+n);
    }
}
```

Syntax of Nested If-else-statement:-

```
if<condition1>
        Statement_blocks1;
   else
        if<condition2>
          Statement_blocks2;
         else
            if<condition3>
                Statement_blocks3;
           else
                if<condition4>
                 Statement_blocks4;
                 …
                …
                …
                else
                 Statement_blocks_Exit;
```

Example:-The largest of any Three Numbers:-

```
        import java.io.*;
class pr11
{
 public static void main(String args[])
 {
    DataInputStream in=new DataInputStream(System.in);
    int a=0,b=0,c=0;
    try
    {
       System.out.println("Enter first number=");
       a=Integer.parseInt(in.readLine());
       System.out.println("Enter second number=");
       b=Integer.parseInt(in.readLine());
        System.out.println("Enter third number=");
        c=Integer.parseInt(in.readLine());
         if(a==b && b==c)
         System.out.println("All numbers are equal="+a);
         else
        if(a>b && b>c || a>c && c>b)
         System.out.println("A is the largest Number="+a);
```

```
        else
          if(b>c && c>a|| b>a && a>c)
        System.out.println("B is the largest Number="+b);
        else
          System.out.println("C is the largest Number="+c);
      }
      catch(Exception Jitendra){}
    }
}
```

Example:-The Grading System:-

```
import java.io.*;
class pr15
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
   double h=0,e=0,m=0,p=0,c=0,tot=0,per=0;
    try
    {
      System.out.println("Enter Marks Obt In Hindi=");
      h=Float.valueOf(in.readLine()).floatValue();
      System.out.println("Enter Marks Obt In English=");
      e=Float.valueOf(in.readLine()).floatValue();
      System.out.println("Enter Marks Obt In Maths=");
      m=Float.valueOf(in.readLine()).floatValue();
      System.out.println("Enter Marks Obt In Physics=");
      p=Float.valueOf(in.readLine()).floatValue();
      System.out.println("Enter Marks Obt In Chemistry=");
      c=Float.valueOf(in.readLine()).floatValue();
      tot=h+e+m+p+c;
      per=tot/5;
      System.out.println(" Hindi="+h+"\t"+" English="+e);
      System.out.println("Maths="+m+"\t"+" Physics="+p);
      System.out.println(" Chemistry="+c);
      System.out.println("Totals="+tot+"\t"+" Percentage="+per);
        if(per>=75 && per<=100 && h>=36 && e>=36 && m>=36 && p>=36&& c>=36)
        System.out.println(" Grade=A & Passed");
        else
       if(per>=65 && per<75 && h>=36 && e>=36 && m>=36 && p>=36&& c>=36)
        System.out.println("Grade= B & Passed ");
        else
          if(per>=45 && per<65 && h>=36 && e>=36 && m>=36 && p>=36&& c>=36)
        System.out.println("Grade= C  & Passed ");
        else
          if(per>=36 && per<45 && h>=36 && e>=36 && m>=36 && p>=36&& c>=36)

        System.out.println("Grade= D  & Passed ");
         else
          if(per<36 && per>0)
        System.out.println("Grade= E & Unsatsfactory ");
         else
        System.out.println("Fail & Again Do Hard Work Trust Upon God!");
      }
      catch(Exception Jitendra){}
```

```
    }
}
```

Else If construct/Ladder Statement:-

It is used for solving choice based problems.

Syntax -else-if construct statement:-

```
Syntax:-
        …
        …
        else
          if<condition>
            Statement_blocks;
                …
                …
                …
```

Example:-Choice based problem:-

```
import java.io.*;
class pr13
{
 public static void main(String args[])
 {
    DataInputStream in=new DataInputStream (System.in);
   double a=0,b=0,sum=0,pro=0,div=0,dif=0;
   int ch=0;
    try
    {
         System.out.println("Arithmetic Operation”);
         System.out.println("-------------------");
         System.out.println("1:-SUM");
         System.out.println("2:-DIFFERENCE”);
         System.out.println("3:-PRODUCT");
        System.out.println("4:-DIVISION");
        System.out.println("-------------------");
         System.out.println("Enter Your Choice between(1-4)"+"\n");
         ch=Integer.parseInt(in.readLine());
         System.out.println("Enter Any Two Numbers"+"\n");
         a=Float.valueOf(in.readLine()).floatValue();
         b=Float.valueOf(in.readLine()).floatValue();
         if(ch==1)
         {
             sum=a+b;
             System.out.println("Sum="+sum+"\n");
         }
         else
         if(ch==2)
         {
            dif=a-b;
            System.out.println("Difference="+dif+"\n");
         }
         else
         if(ch==3)
         {
            pro=a*b;
           System.out.println("Product="+pro+"\n");
         }
         else
         if(ch==4)
```

```
        {
          div=a/b;
          System.out.println("Div="+div+"\n");
        }
        else
      System.out.println("Wrong Choice===!Dileep"+"\n");
      }
      catch(Exception ICSM){ }
    }
}
```

Syntax switch-case statement :-

It is also used for making choice based problems.

Syntax:-

```
switch(expression)
{
 case <value1>:
        Statement_Blocks;
        break;
case <value2>:
        Statement_Blocks;
        break;
case <value3>:
        Statement_Blocks;
        break;
case <value4>:
        Statement_Blocks;
        break;
        …
        …
        …
default:
Statement_Blocks_False;
}
Exit_Statement;
```

Example:- Conversion of Numbers into Words

```
        import java.io.*;
class pr14
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
    int n=0;
    try
    {
        System.out.println("Conversion of Number Into Words="+"\n");
        System.out.println("-----------------");
        System.out.println("Enter any Number to Convert Into words"+"\n");
        n=Integer.parseInt(in.readLine());
        switch(n)
        {
          case 1:
```

```
            System.out.println("One"+"\n");
            break;
             case 2:
            System.out.println("Two"+"\n");
            break;
             case 3:
            System.out.println("Three"+"\n");
            break;
             case 4:
            System.out.println("Four"+"\n");
            break;
             case 5:
            System.out.println("Five"+"\n");
            break;
             case 6:
            System.out.println("Six"+"\n");
            break;
             case 7:
            System.out.println("Seven"+"\n");
            break;
             case 8:
            System.out.println("Eight"+"\n");
            break;
             case 9:
            System.out.println("Nine"+"\n");
            break;
          default :
           System.out.println("Wrong Number"+"\n");
        }
    }
        catch(Exception Sandhya){}
    }
}
```

LECTURE - 4

Looping Control Statement/Iteration Control/Repition Control:-

Syntax of While loop:-

It executes looping body when condition is true.And exit from loop when condition is false.

```
            …
            …
            …
        while(Criteria)
            {
            Statement_Body_True_Condition;
            Updations;
            }
            Exit_Statement_False_criteria;
```

Example:-15

```
import java.io.*;
class pr15
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
    int n=0,i=1,s=0;
    try
    {
          System.out.println("Series of Natural numbers and their Sum="+"\n");
          System.out.println("-----------------");
          System.out.println("Enter Any Number"+"\n");
           n=Integer.parseInt(in.readLine());
          while(i<=n)
          {
            System.out.println(i+"\t");
                    s=s+i;
                    i++;
          }
                    System.out.println("\n\nSum="+s);
          }
       catch(Exception Sanjay){}
    }
}
```

Example:-16

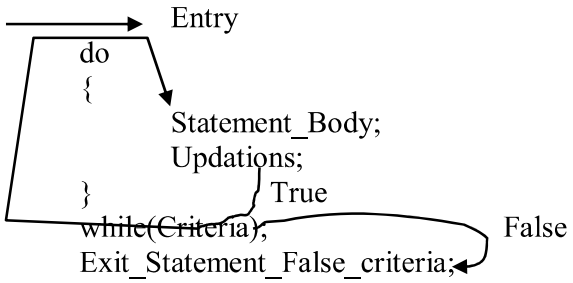
Armstrong Number:-

```
import java.io.*;
class pr16
{
 public static void main(String args[])
 {
    DataInputStream in=new DataInputStream (System.in);
    int n=0,m=0,d=0,s=0,a=1,k=0;
    try
    {
          System.out.println("Series of Armstrong Numbers="+"\n");
           System.out.println("Enter Any Number"+"\n");
           k=Integer.parseInt(in.readLine());
           while(a<=k)
           {
                    n=a;
                    s=0;
                    while(n>0)
                    {
                    m=n%10;
                    n=n/10;
                    s=s+m*m*m;
                    }
                    if(s==a)
                    System.out.println("Series of Armstrong Numbers="+a+"\n");
                    a++;
                    }
```

```
            }
        catch(Exception Sandhya){}
    }
}
```

Syntax of do-While loop:-

It executes looping body at least one time when condition is true or false.
And execute further when condition is true.



Example:-17

```
import java.io.*;
class pr17
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
    int n=0,i=1,i1=2,s1=0,s2=0;
    try
    {
        System.out.println("Series of Odd And Even numbers and their Sums="+"\n");
        System.out.println("--------------------");
        System.out.println("Enter Any Number"+"\n");
         n=Integer.parseInt(in.readLine());
        System.out.println("Odd Number Series=");
  do
{
   System.out.println(i);
   i=i+2;
   s1=s1+i;
}while(i<=n);

System.out.println("Sum of Odd Numbers="+s1);
System.out.println("Even Number Series=");
do
{
   System.out.println(i1);
 i1=i1+2;
s2=s2+i1;
}while(i1<=n);
System.out.println ("Sum of Even Numbers="+s2);
}
      catch(Exception Sameer){}
   }
}
```

Example:-18

```
import java.io.*;
```

```
class pr18
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
    int n=0,i=1,a=0,b=1,s=0;
    try
    {
          System.out.println("Series of Fibonacci Series="+"\n");
      System.out.println("--------------------");
          System.out.println("Enter Any Number"+"\n");
           n=Integer.parseInt(in.readLine());
do
{
 System.out.print(s+"\t");
a=b;
b=s;
s=a+b;
i++;
}while(i<=n);
 }
       catch(Exception Sandhya){}
    }
}
```

For loop:-It is useful while executing a statement a fixed number of times.Even here,More than one statement can be enclosed in curly braces to form a compound statement.

Syntax of for loop:-

```
for(initializations;Condition; Updations)
          Statement_Blocks;
          Exit_Statement;
```

True

False

Intializations:-There are many initializations by using commas.
Updations:-There are many Updations by using commas.
Condition:-Only one Condition will be defined.

Example:-19

```
import java.io.*;
class pr19
{
 public static void main(String args[])
 {
    DataInputStream in=new DataInputStream(System.in);
    int i=0,n;
    try
    {
          System.out.println("Series of ICSM="+"\n");
          System.out.println("-------------------");
          System.out.println("Enter Any Number"+"\n");
           n=Integer.parseInt(in.readLine());
           for (i=1;i<=n;i++)
           System.out.print ("ICSM"+"\t");
```

```
        }
      catch (Exception Sandhya){}
   }
}
```

Example:-20

```
import java.io.*;
class pr20
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
    int i=1,n,fact=1;
    try
    {
         System.out.println("Series of Factorial Numbers="+"\n");
         System.out.println("-------------------");
         System.out.println("Enter Any Number"+"\n");
          n=Integer.parseInt(in.readLine());
          for(i=1;i<=n;i++)
          {
          fact=fact*i;
          System.out.println("Factorial of Number="+i+"="+fact+"\t");
         }
         }
      catch(Exception Sandhya){}
   }
}
```

Example:-21

```
import java.io.*;
class pr21
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
    double  i=1,n,m=0;
    try
    {
         System.out.println("Series of Power Numbers="+"\n");
         System.out.println("-------------------");
         System.out.println("Enter Any Number"+"\n");
          n=Double.parseDouble(in.readLine());
          for(i=1;i<=n;i++)
          {
           m=Math.pow(i,i);
          System.out.println("Series of Power="+i+"="+m+"\t");
         }
         }
      catch(Exception Sandhya){}
   }
}
```

Example:-22

```
import java.io.*;
```

```
class pr22
{
 public static void main(String args[])
 {
    DataInputStream in=new DataInputStream(System.in);
    int i=1,j=0,n=0;
    try
    {
          System.out.println("Example based On Nested For Loop="+"\n");
          System.out.println("-------------------");
          System.out.println("Enter Any Number"+"\n");
           n=Integer.parseInt(in.readLine());
           for(i=1;i<=n;i++)
           {
            for(j=i;j<=n;j++)
           System.out.print(i);
           System.out.print("\n");
          }
          }
       catch(Exception Sandhya){}
    }
}
```

Example:-23

```
import java.io.*;
class pr23
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream (System.in);
    int i=1,j=0,n=0;
    try
    {
          System.out.println ("Example based On Nested For Loop="+"\n");
          System.out.println ("-------------------");
          System.out.println ("Enter Any Number"+"\n");
           n=Integer.parseInt(in.readLine());

           for(i=1;i<=n;i++)
           {
            for(j=i;j<=n;j++)
           System.out.print(j);
           System.out.print("\n");
          }
          for(i=1;i<=n;i++)
           {
            for(j=i;j<=n;j++)
           System.out.print(" ☻ ");
           System.out.print("\n");
          }
          for(i=1;i<=n;i++)
           {
            for(j=i;j<=n;j++)
           System.out.print("*");
           System.out.print("\n");
```

```
        }
        }
      catch(Exception Sandhya){}
    }
}
```

Continue Control Statement:-

This statement skips the remainder of the current iteration and initiates the execution of the next iteration. When this statement is encountered in a loop, the rest of the statements in the loop are skipped and the control passes to the condition, which is evaluated, and if true, the loop is entered again.

Syntax:-

```
continue;
```

Example:-

```
class pr31
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream (System.in);
    int i=1,j=0,n=0;
    try
    {
        System.out.println ("Example based On continue="+"\n");
        System.out.println ("--------------------");
        System.out.println ("Enter Any Number"+"\n");
         n=Integer.parseInt(in.readLine());

         for(i=1;i<=n;i++)
         {
          if(i%2==0)
        continue;
         System.out.println (i);
        }
      catch(Exception s1){}
    }
}
```

Note:-break statement used only in switch case control and looping control statement.

Syntax:-

```
break;
```

LECTURE - 5

Array:-

Collection of similar data types elements is called array.

Syntax for Single dimensional Array:-

```
data_Type Array_name[]=new data_Type[size];
```

Syntax for Double dimensional Array:-

```
data_Type Array_name[][]=new data_Type[size1] [size2];
```

Example:-1

```
int m[]=new int [10];
```

m[0]	m[1]	m[2]	m[3]	m[4]	m[5]	m[6]	m[7]	m[8]	m[9]
4	8	7	9	9	5	2	6	4	1

Example:-2
int m[][]=new int [3][3];

m[0][0]	m[0][1]	m[0][2]
m[1][0]	m[1][1]	m[1][2]
m[2][0]	m[2][1]	m[2][2]

12	15	19
18	19	14
28	26	35

Example:-24

```
import java.io.*;
class pr24
{
 public static void main(String args[])
 {
    DataInputStream in=new DataInputStream(System.in);
   int m[]=new int[10],i=0;
    try
    {
         System.out.println("Enter Elements in Single dimensional Array="+"\n");
        System.out.println("-------------------");
        for(i=0;i<=9;i++)
        m[i]=Integer.parseInt(in.readLine());
        System.out.println("-------------------");
        System.out.println("Entered Elements in Single dimensional Array="+"\n");
        for(i=0;i<=9;i++)
         System.out.print(m[i]+"\t");
      }
       catch(Exception Sandhya){}
   }
}
```

Example:-25

```
import java.io.*;
class pr26
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
   int m[][]=new int[3][3],row=0,col=0;

    try
    {
         System.out.println("Enter Elements Matrix="+"\n");
        System.out.println("--------------------");
        for(row=0;row<=2;row++)
         for(col=0;col<=2;col++)
         m[row][col]=Integer.parseInt(in.readLine());
```

```
        System.out.println("-------------------");
        System.out.println("Matrix=="+"\n");
       for(row=0;row<=2;row++)
       {
         for(col=0;col<=2;col++)
         System.out.print(m[row][col]+"\t");
          System.out.print("\n");
         }

          System.out.println("Transpose Matrix=="+"\n");
       for(row=0;row<=2;row++)
       {
         for(col=0;col<=2;col++)
         System.out.print(m[col][row]+"\t");
          System.out.print("\n");
         }
      }
       catch(Exception c1){}
   }
}
```

Example:-26

```
import java.io.*;
class pr27
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
   int m1[][]=new int[3][3];
   int m2[][]=new int[3][3];
   int m3[][]=new int[3][3],row=0,col=0;
    try
    {
         System.out.println("Enter Elements Matrix1="+"\n");
        System.out.println("-------------------");
        for(row=0;row<=2;row++)
         for(col=0;col<=2;col++)
         m1[row][col]=Integer.parseInt(in.readLine());
         System.out.println("Enter Elements Matrix2="+"\n");
        System.out.println("-------------------");
        for(row=0;row<=2;row++)
         for(col=0;col<=2;col++)
         m2[row][col]=Integer.parseInt(in.readLine());
        System.out.println("--------------------");
        System.out.println("Matrix1=="+"\n");
       for(row=0;row<=2;row++)
       {
         for(col=0;col<=2;col++)
         System.out.print(m1[row][col]+"\t");
          System.out.print("\n");
         }
          System.out.println("Matrix2=="+"\n");
       for(row=0;row<=2;row++)
       {
         for(col=0;col<=2;col++)
```

```
          System.out.print(m2[row][col]+"\t");
           System.out.print("\n");
          }

          System.out.println("Sum of Two Matrix=="+"\n");
        for(row=0;row<=2;row++)
        {
           for(col=0;col<=2;col++)
          m3[row][col]= m1[row][col]+ m2[row][col];

          }
           for(row=0;row<=2;row++)
        {
          for(col=0;col<=2;col++)
          System.out.print(m3[row][col]+"\t");
           System.out.print("\n");
          }


      }
        catch(Exception gaurav){}
   }
}
```

Selection Sort:-

Example:-27 (Sorting of Unordered Elements in Descending Order)

```
import java.io.*;
class pr28
{
 public static void main(String args[])
 {

    DataInputStream in=new DataInputStream(System.in);
   int m[]=new int[10],i=0,j=0,t=0;

    try
    {
        System.out.println("Enter Unordered Elements in Single dimensional Array="+"\n");
        System.out.println("--------------------");
        for(i=0;i<=9;i++)
        m[i]=Integer.parseInt(in.readLine());
        System.out.print("Enter Elements in Single dimensional Array="+"\n");
        for(i=0;i<=9;i++)
        System.out.print(m[i]+"\t");
        System.out.println("\nOrdered Elements in Single dimensional Array in Descending Order="+"\n");
          for(i=0;i<=9;i++)
          {
              for(j=0;j<=9;j++)
              {
              if(m[i]>m[j])
              {
                 t=m[i];
                 m[i]=m[j];
                 m[j]=t;
               }
            }
          }
```

```
            for(i=0;i<=9;i++)
           System.out.println(m[i]+"\t");

        }
          catch(Exception Sandhya){}

    }
}
```

Bubble_Sort:-

```
import java.util.Scanner;
class bubble_sort
 {
  public static void main()
{
    int n, c, d, swap;
    Scanner in = new Scanner(System.in);
    System.out.println("Input number of integers to sort");
    n = in.nextInt();
    int array[] = new int[n];
    System.out.println("Enter " + n + " integers");
    for (c = 0; c < n; c++)
      array[c] = in.nextInt();
    for (c = 0; c < ( n - 1 ); c++) {
      for (d = 0; d < n - c - 1; d++) {
        if (array[d] > array[d+1])
        {
          swap       = array[d];
          array[d]   = array[d+1];
          array[d+1] = swap;
        }
      }
    }
    System.out.println("Sorted list of numbers");
    for (c = 0; c < n; c++)
      System.out.print(array[c]+"\t");
  }
}
```

LECTURE - 6

String:- **Important**

It is sequence of characters enclosed within double quote.

Example:-

"VARANASI" Consist of 9 characters

'V','A','R','A','N','A','S','I','\0'

Note:- String terminated by null character.

Syntax:-

```
String <array_Name>[]=new String[size];
                Or
String  <String_Name>;
<String_Name>=new String ("String Name");
```

Example:-28 (Displaying of declared string):-

```
import java.io.*;
class pr29
{
 public static void main(String args[])
 {
    String str1;
    str1=new String("WELCOME");
    System.out.println(str1);
   }
}
```

How to input string through through keyboard:-

```
import java.io.*;
public class str2
{
  public static void main(String args[]) throws IOException
    {
       InputStreamReader input=new InputStreamReader(System.in);
      BufferedReader keyboardInput=new BufferedReader(input);
       String name,fname,mname,cname;
       System.out.println("Enter name of student=");
       name = keyboardInput.readLine();
       System.out.println("Enter name of father=");
       fname = keyboardInput.readLine();
        System.out.print(fname);
        System.out.println("Enter name of mname=");
        mname = keyboardInput.readLine();
        System.out.println("Enter name of course=");
       cname = keyboardInput.readLine();
       System.out.print(cname);
         System.out.println("Name of student="+name);
        System.out.println("Name of Father="+fname);
        System.out.println("Name of Mother="+mname);
   }
}
```

Function Related String:-

Let str1 and str2 are two strings.

1. For uppercase
   a. s2=str1.toUpperCase();
2. For Lowercase
   a. s2=str1.toLowerCase();
3. For Concatenation
   a. str1.concat(str2);
4. For equal of two string
   a. str1.equals(str2);
5. For calculating length of string
   a. str1.length();
6. For Removing unwanted spaces from beginning and end of string
   a. str2=str1.trim();
7. For Replace all x with y
   a. str2=str1.replace('x','y');

8. For give nth characters of str1
    a. str2=str1.ChartAt(n);
9. For Gives substring starting from n^{th} character up to m^{th}
    a. str2=str1.substring(n,m);
10. For Gives substring starting from n^{th} character
    a. str2=str1.substring(n);

Example:-29 (Displaying of string result using string function):-

```
import java.io.*;
class pr29
{
 public static void main(String args[])
 {
    String str1,str2,str3;
    int m1=0,m2=0;
    str1=new String("WELCOME");
    str2=new String("VARANASI");
      m1=str1.length();
      m2=str2.length();
    System.out.println("Length Of String str1="+m1);
    System.out.println("Length Of String str2="+m2);
   }
}
```

Example:-30 (Displaying of string result using string function):-

```
import java.io.*;
class pr30
{
 public static void main(String args[])
 {
    String str1,str2,str3;
    int m1=0,m2=0;
    str1=new String("WELCOME");
    str2=new String("\tVARANASI");
     str3=str1.concat(str2);
    System.out.println("String after Concatenation="+str3);

   }
}
```

LECTURE - 7

Problem Based On using Class and Object:-

How To Define Class , Object and Access Method:-

```
class <class_Name>
{
        ----
        ----
public:
        Data_members;
        Member_functions;
```

```
---
---

};
```

Creation of Object:-

```
<class_Name>  <object_name>;
<object_name> =new  <class_Name>( );
```

Access Method:-

```
<object_name>.<function_name>();
```

Example:-31

```
import java.io.*;
class pr31
{
public static void main(String args[])
{
base1 obj1;
obj1=new base1();
base2 obj2;
obj2=new base2();
obj1.f1();
obj2.f2();
}
}
class base1
{
public static void f1()
{
System.out.println("Member of Class Base1");
}
};
class base2
{

public static void f2()
{
System.out.println("Member of Class Base2");
}
};
```

Example:-32(Passing No Argument with no return value in function)

```
import java.io.*;
class pr32
{
public static void main(String args[])
{
base1 obj1;
obj1=new base1();
obj1.f1();
}
}
class base1
{
public static void f1()
```

```
{
 int a=5,b=12,c=0,c1=0;
 c=a+b;
 c1=a*b;
System.out.println("Sum of  Two Numbers=" +c);
System.out.println("Product of  Two Numbers=" +c1);
}
};
```

Example:-33 (Passing Argument with return value in function)

```
import java.io.*;
class pr33
{
public static void main(String args[])
{
   int p1;
  base1 obj1;
  obj1=new base1();
  p1=obj1.product();
  obj1.x=12;
  obj1.y=15;
  p1=obj1.product();
  System.out.println("Product="+p1);
}
}
class base1
{
 int x,y;
 void f1(int a,int b)
{
x=a;
y=b;
}
int product()
{
int m;
m=x*y;
return (m);
}
};
```

Example:-34(Passing Argument with return value but input Through keyboard)

```
import java.io.*;
class pr34
{
public static void main(String args[])
{
DataInputStream in=new DataInputStream(System.in);
 int p1,p2,m1=0,n1=0;
 try
  {

base1 obj1;
obj1=new base1();
p1=obj1.product();

base2 obj2;
```

```
obj2=new base2();
p2=obj2.sum();
System.out.println("Enter First Number=");
 n1=Integer.parseInt(in.readLine());
System.out.println("Enter Second Number=");
 m1=Integer.parseInt(in.readLine());
obj1.x=m1;
obj1.y=n1;
p1=obj1.product();
obj2.x=m1;
obj2.y=n1;
p2=obj2.sum();
System.out.println("Product="+p1);
System.out.println("Sum="+p2);
}
 catch(Exception Sandhya){}
}
}
class base1
{
 int x,y;
 void f1(int a,int b)
{
x=a;
y=b;
}
int product()
{
int m;
m=x*y;
return (m);
}
};

class base2
{
 int x,y;
 void f2(int a,int b)
{
x=a;
y=b;
}
int sum()
{
int p;
p=x+y;
return (p);
}
};
```

Example:-35(Program of Leap year using OOPs)

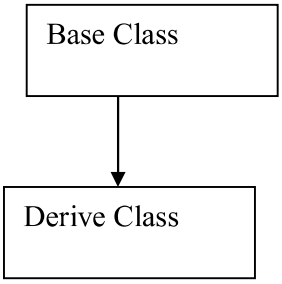
```
import java.io.*;
class pr35
{
public static void main(String args[])
{
```

```
DataInputStream in=new DataInputStream(System.in);
 int p1,year=0;
 try
 {

base1 obj1;
obj1=new base1();
p1=obj1.leap();

System.out.println("Enter Any Year Number=");
 year=Integer.parseInt(in.readLine());
obj1.y=year;
p1=obj1.leap();
if(p1==0)
 System.out.println("Leap Year");
 else
 System.out.println("Not Leap Year");
}
 catch(Exception Sweta){}
}
}
class base1
{
 int y;
 void in_year(int year1)
{
y=year1;
}
int leap()
{
int m;
m=y%4;
return m;
}
};
```

LECTURE - 8

Inheritence:- (Important)

It means reusability.That is property of base class is going to access by its derive class. Whose property is going access is called as base class and in which the property is access is called as a derive class.Inheritence performed in java by using keywords **extends.**



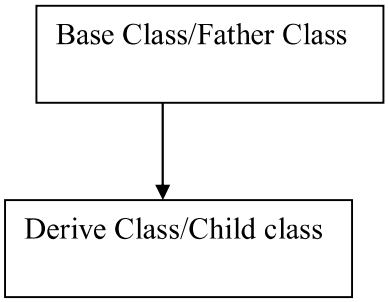
Syntax:-

```
class <deriveclass> extends <base_class>
{
 Variable_Declarations;
 Methods_Declarations;
```

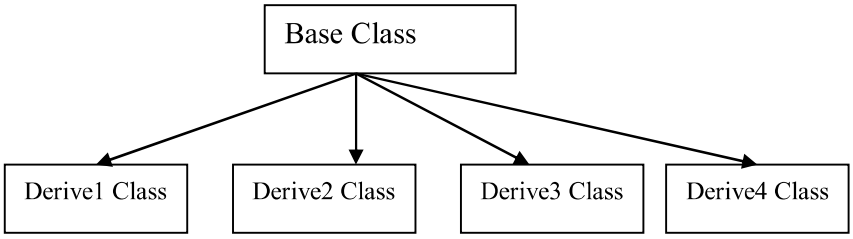
}

Forms of Inheritence:-

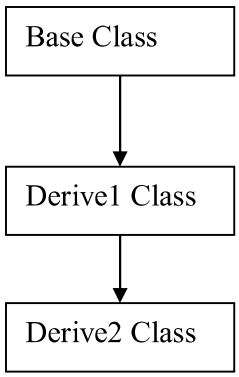
Single Inheritence



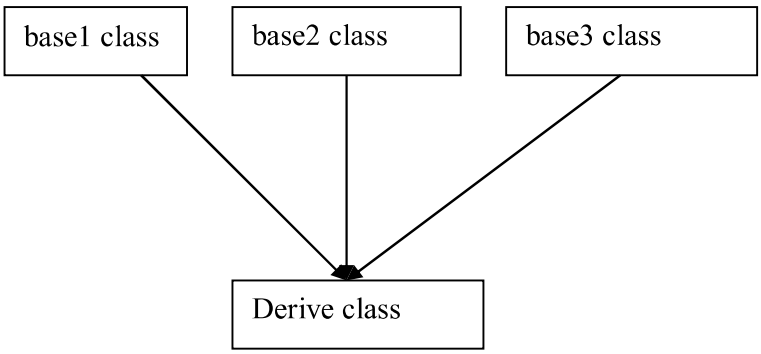
Hierarchical Inheritence



Multilevel Inheritence



Multiple Inheritence



Example of Single Inheritence:-

Example:-36

```
import java.io.*;
class pr36
{
public static void main(String args[])
{
base1 obj1;
obj1=new base1();
derive obj2;
obj2=new derive();
obj1.f1();
obj2.f2();
obj2.f1();
}
};
class base1
{
public static void f1()
{
System.out.println("Member of Class Base1");
}
};
class derive extends base1
{

public static void f2()
{
System.out.println("Member of Class Derive");
}
};
```

Example of Hierarchical Inheritance:-

Example:-37

```
import java.io.*;
class pr37
{
public static void main(String args[])
{
base1 obj1;
obj1=new base1();
derive1 obj2;
obj2=new derive1();
derive2 obj3;
obj3=new derive2();
derive3 obj4;
obj4=new derive3();
obj1.f1();
obj2.f2();
obj3.f3();
obj4.f4();
obj2.f1();
obj3.f1();
obj4.f1();
}
```

```
}
class base1
{
public static void f1()
{
System.out.println("Member of Class Base1");
}
};
class derive1 extends base1
{
public static void f2()
{
System.out.println("Member of Class Derive1");
}
};
class derive2 extends base1
{
public static void f3()
{
System.out.println("Member of Class Derive2");
}
};
class derive3 extends base1
{
public static void f4()
{
System.out.println("Member of Class Derive3");
}
};
```

Example of Multilevel Inheritance:-

Example:-38

```
import java.io.*;
class pr38
{
public static void main(String args[])
{
base1 obj1;
obj1=new base1();
derive1 obj2;
obj2=new derive1();
derive2 obj3;
obj3=new derive2();
derive3 obj4;
obj4=new derive3();
obj1.f1();
obj2.f2();
obj3.f3();
obj4.f4();
obj4.f1();
obj4.f2();
obj4.f3();
}
}
class base1
{
```

```
public static void f1()
{
System.out.println("Member of Class Base1");
}
};
class derive1 extends base1
{
public static void f2()
{
System.out.println("Member of Class Derive1");
}
};
class derive2 extends derive1
{
public static void f3()
{
System.out.println("Member of Class Derive2");
}
};
class derive3 extends derive2
{
public static void f4()
{
System.out.println("Member of Class Derive3");
}
};
```

LECTURE - 9

Interfaces:Multiple Inheritence :- Most Important

An interface is basically a kind of class.An interface contain methods and variables.It define abstract methods and final fields.It means that interfaces do not specify any code to implement these methods and fields contain only constants.

Syntax:-

```
interface <interface_name1>
{
Variable_Declarations;
Method_Declarations;
}
```

How to Extends interfaces

```
interface <interface_name2> extends <interface_name1>
{
Variable_Declarations;
Method_Declarations;
}
```

Implementing Interfaces in class

Syntax1:-

```
class <class_Name> implements <interface_name1>
{
Variable_Declarations;
Method_Declarations;
}
```

Syntax2:-

```
class <class_Name> extends <Superclass> implements <interface_name1>,
<interface_name2>, < interface_name3>,…
```

```
{
        Variable_Declarations;
        Method_Declarations;
}
```

Variables declarations:-

```
static final <data_type> <variable_name>=<"Value">;
```

Example:- 39

```
import java.io.*;
class inter
{
 public static void main(String args[])
   {

    base obj1=new base( );
    Calculation obj2;
    obj2=obj1;
    System.out.println("Calculation of Two Numbers="+obj2.cal(12,50));
   }
}
interface Calculation
{
   int cal(int x,int y);
 }
 class base implements Calculation
 {
   public int cal(int x,int y)
   {
     return x+y;
   }
}
```

Example:- 40

```
import java.util.Scanner;
import java.io.*;
class inter
{
 public static void main(String args[])
   {
    base obj1=new base();
    Calculation1 obj2;
    obj2=obj1;
   int a,b;
  Scanner sc=new Scanner(System.in);
  System.out.println("Enter Value of n=");
  a=sc.nextInt();
  System.out.println("Enter Value of m=");
  b=sc.nextInt();
  System.out.println("Product  of Two Numbers="+obj2.cal(a,b));
   }
}

interface Calculation
{
```

```
    int cal(int x,int y);
  }
  class base implements Calculation
  {
    public int cal(int x,int y)
    {
      return x*y;
    }
}
```

Example:- 41

```
import java.io.*;
class inter2
{
 public static void main(String args[])
    {
     DataInputStream in=new DataInputStream(System.in);
     int n1=0,n2=0;
     base obj1=new base();
     Calculation cal_obj;
     cal_obj=obj1;
     try
     {
     System.out.println("Enter First Number"+"\n");
      n1=Integer.parseInt(in.readLine());
      System.out.println("Enter Second Number"+"\n");
      n2=Integer.parseInt(in.readLine());
    }
     catch(Exception Sandhya){}
     System.out.println("Calculation of Two Numbers="+cal_obj.cal(n1,n2));
    }
}
interface Calculation
{
    int cal(int x,int y);
  }
  class base implements Calculation
  {
    public int cal(int x,int y)
    {
      return x+y;
    }
}
```

Example:- 42

```
import java.io.*;
class inter3
{
 public static void main(String args[])
    {
     base obj1=new base();
     inter1 inter1_obj;
     inter1_obj=obj1;
     inter1_obj.f1();
     base1 obj2=new base1();
     inter2 inter2_obj;
```

```
    inter2_obj=obj2;
    inter2_obj.f2();
   }
}
interface inter1
{
     void f1();
 }
 interface inter2
{
     void f2();
 }
 class base implements inter1
 {
   public void f1()
   {
      System.out.println("Welcome1");
    }
}
class base1 extends base implements inter1,inter2
 {
   public void f2()
   {
      System.out.println("Welcome2");
    }
}
```

LECTURE - 10

Multithreaded Programming:- /Example of Polymorphism Most Important

It is conceptual programming paradigm where a program is divided into two or more subprograms, which can be implemented at the same time in parallel. It is unique property of java.

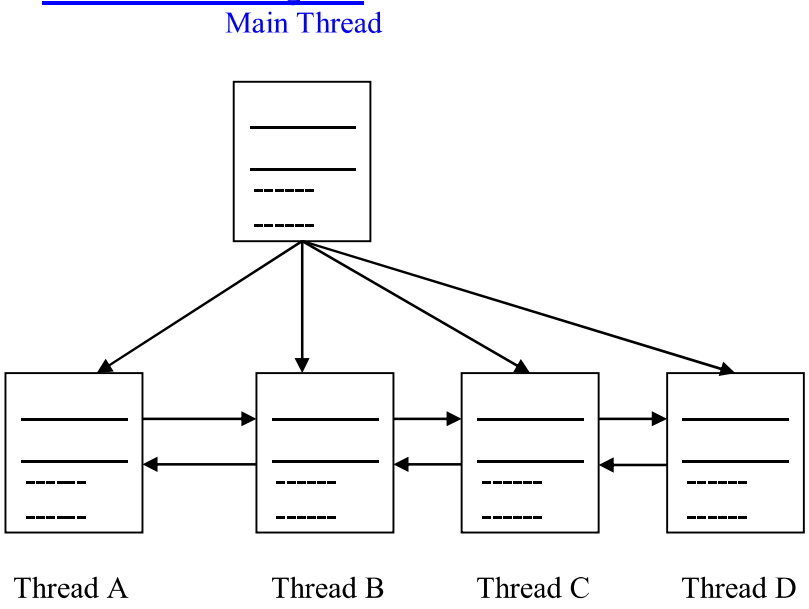
Note:-Thread is a process

- Single Threaded Program.
- Multithreaded Program.

Single Threaded Program

```
Class <class_Name>
{
…            Beginning
…
…
…            Single Threaded  body of execution
…
…            End
}
```

Multithreaded Program.



Syntax:-

```
public void run()
{
……..
……..
……..
}
```

Extending The Thread Class:-

It can be extended as follows:-

```
class <thread_Nmae> extends Thread

{
…
…
…
}
```

Example:- 43

```
import java.io.*;
class thr1
{
public static void main(String args[])
{
new base_thread1().start();
new base_thread2().start();
new base_thread3().start();
}
}
class base_thread1 extends Thread
{
 public void run()
  {
```

```
            int a1=5,b1=7,c1;
            c1=a1+b1;
        System.out.println("Welcome In Thread1"+"C1="+c1);
        }
      }
      class base_thread2 extends Thread
      {
       public void run()
       {
            int a2=15,b2=17,c2;
            c2=a2*b2;
        System.out.println("Welcome In Thread2"+"C2="+c2);
       }
      }
      class base_thread3 extends Thread
      {
       public void run()
       {
             float a3=15,b3=17,c3;
             c3=a3/b3;
        System.out.println("Welcome In Thread3"+"C3="+c3);
       }
      }
```

Stopping and Blocking A thread:-

Syntax Name:-

```
<Thread_Name>.stop();
```

Example:- 44

```
import java.io.*;
class thr1
{
public static void main(String args[])
{

    new base_thread1().start();

}
}
class base_thread1 extends Thread
{
 public void run()
  {
     int i,n;
     for(i=1,n=5;i<=n;i++)
     {
      System.out.println("Welcome In Thread1");
      if(i==3)
      stop();
     }
  }
}
```

Example:- 45

sleep ( ) Blocked for a specified time:-

```
import java.io.*;
class thr1
{
public static void main(String args[])
{

    new base_thread1().start();

}
}
class base_thread1 extends Thread
{
 public void run()
  {
     int i,n;
     for(i=1,n=5;i<=n;i++)
     {
        try
        {

           sleep(2000);
           System.out.println("Welcome In Thread1");

          }
       catch(Exception m1){}
    }
  }
}
```

suspend( ) Blocked until further orders:-

Example:- 46

```
import java.io.*;
class thr1
{
public static void main(String args[])
{
    new base_thread1().start();
}
}
class base_thread1 extends Thread
{
 public void run()
  {
     int i,n;
     for(i=1,n=5;i<=n; i++)
     {
        System.out.println("Welcome In Thread1");
         if(i>=3)
         suspend();
    }
  }
}
```

wait( ) **Blocked until certain condition occures:-**

Example:- 47

```
                    import java.io.*;
class thr1
{
public static void main(String args[])
{
    new base_thread1().start();
}
}
class base_thread1 extends Thread
{
 public void run()
  {
     int i,n;
     for(i=1,n=5;i<=n; i++)
     {
        if(i<=3)
        try
        {
         wait(2000);
         System.out.println("Welcome In Thread1");
        }
       catch(Exception m1){}
    }
  }
}
```

LECTURE - 11

Applet Programming:-

Applets are small java programs that are primarily used in internet computing.They can be transported over the internet from one computer to another and run using the applet viewer or any web browser that supports java.

Building of applet tag:-

```
import java.awt.*;
public class <class_name> extends Applet
 {
        public void main paint(Graphics <object_name>)
        {
        < object_Name>.function_name(arg1,arg2,arg3…);
        }
}
```

Applet tag:-

```
<applet  code="javafile.class" width="Values in Pixels" Height=" Values in Pixels">

</applet>
```

Adding Applet To HTML:-

```
<HTML>
<Head>…</head>
<Title>…</title>
<body>
….
<applet code="javafile.class"
                width="Values in Pixels"
                Height="Values in Pixels">
</applet>
….
</body>
</html>
```

Example:- 48

```
import java.awt.*;
import javax.swing.*;
public class apl1 extends JApplet
{
   public void paint(Graphics g)
   {
      g.setColor(Color.red);
      g.drawString("ICSM", 20, 20);
      g.setColor(Color.blue);
      g.drawString("ICSM IS A TAPOBHUMI", 20, 40);
   }
}
```

Example:- 49

```
import java.awt.*;
import javax.swing.*;
public class apl1 extends JApplet
{
   public void paint(Graphics g)
   {
      g.setColor(Color.red);
      g.fillRect(30,40,80,120);
      g.drawString("ICSM IS A TAPOBHUMI", 20, 40);
   }
}
```

Example:- 50

```
import java.awt.*;
import javax.swing.*;
public class apl3 extends JApplet
{
   public void paint(Graphics g)
   {
      g.setColor(Color.red);
      g.drawLine(20,30,250,40);
      g.setColor(Color.green);
      g.fillRoundRect(30,50,200,50,5,5);
      g.setColor(Color.blue);
      g.fillRoundRect(60,90,400,90,15,15);
   }
}
```

Example:- 51

```
import java.awt.*;
import javax.swing.*;
public class apl3 extends JApplet
{
   public void paint(Graphics g)
   {
      g.setColor(Color.red);
      g.drawLine(20,30,250,40);
      g.setColor(Color.green);
      g.fillRoundRect(30,50,200,50,5,5);
      g.setColor(Color.blue);
      g.drawOval(90,130,40,80);
      g.setColor(Color.green);
      g.fillOval(120,130,40,80);
   }
}
```

LECTURE - 12

Introduction Of Package:- Most important

Packages in java provide a way of grouping variety of classes and/or interfaces together.The grouping is usually done according to functionality.It includes following benefits.

- The classes contained in the packages of other programs can be easily reused.
- In packages, classes can be unique compared with classes in other packages.
- It provides a way to "hide".
- It also provides a way for separating design from coding.

Java API Packages:-

It provides a large numbers of classes grouped into different packages according to the functionality. Some of most useful packages available with java API.

- Java.lang — It supports language classes. That includes Strings, maths, exceptions and threads.
- Java.util — It includes utility classes of languages that may support random numbers, date, hash tables etc.
- java.io — It support input /output classes.
- Java.awt — It includes classes for windows, button, lists, menus and so on.
- java.net — It support networking.
- java.applet — It is used for creating and implementing applets.

Syntax for creating package:-

```
package <package_name>
{
…….
…….   {Body of class}
…….
}
```

How to access a package:-

```
import packag1[.package2] [.package3].<class_name>;
```

Package1

```
public class pkg1
{
public void sum()
{
int a=6,b=9,p1;
p1=a+b;
```

```
System.out.println("Sum="+p1);
}
}
```

Package2

```
public class pkg2
{
public void pro()
{
int a=6,b=9,p2;
p2=a*b;
System.out.println("Product="+p2);
}
}
```

Package Access

```
import java.io.*;
class prg_pkg1
{
  public static void main()
  {
 pkg1 obj1=new pkg1();
 obj1.sum();
 pkg2 obj2=new pkg2();
 obj2.pro();
}
}
```

Example:- 53(With using arguments inside package function):-

Package:-

```
public class pkg3
{
 int x,y;
 void f1(int a,int b)
{
x=a;
y=b;
}
void product()
{
int m;
m=x*y;
System.out.println("Pro1="+m);
}
}
```

Access Package:-

```
import java.io.*;
   class prg_pkg2
{
   public static void main(String args[])
   {
    DataInputStream in=new DataInputStream(System.in);
    int m1=0,n1=0;
    pkg3 obj3=new pkg3();
    try
    {
    System.out.println("Enter First Number=");
    n1=Integer.parseInt(in.readLine());
    System.out.println("Enter Second Number=");
```

```
    m1=Integer.parseInt(in.readLine());
    obj3.x=m1;
    obj3.y=n1;
    obj3.product();
    }
    catch(Exception Sandhya){}
   }
}
```

LECTURE - 13

Managing Errors(bugs) and Exceptions:-

An error may produce an incorrect output or may terminate the execution of the program abruptly or even may cause the system to crash.

Types of errors:-

- Compile Time errors:-Such types of errors detected by the java compiler and therefore these errors are known as compile time error.
- Run time Errors:-A program may compile successfully creating the .class (Byte code) files but may not run properly.Such program may produce wrong result.

Cause of Compile Time errors:-

- Missing semicolons.
- Missing brackets.
- Misspelling of identifiers and keywords.
- Use of undeclared variables.
- Bad references to objects.
- Use of =(Assignment operator) in place of = = (Equal Operator).

Cause of Run Time errors:-

- Dividing any number by zero (m/0).
- Accessing an element that is out of bounds of an array.
- Trying to store a value into an array of an incompatible class or type.
- Converting invalid string into numbers.
- Passing a parameter that is not in a valid range or value for a method.
- Accessing a character that is not out of bounds of string.

Exception:-

An exception is a condition that is caused by a run time errors in the program when the java interpreter encounters an errors such as dividing any real number by zero.

Java Exception Class:-

- ArithmeticException. Caused by math errors such as divide by zero.
- ArrayIndexOutOfBoundsException. Caused by bad array indexes.
- IOException. Caused by general I/O failure.
- ArrayStoreException. Caused when a program tries to store the wrong type data in an array

Example:- 54 (Based on Exception)

```
import java.io.*;
  class prg_pkg2
{
   public static void main(String args[])
   {
    DataInputStream in=new DataInputStream(System.in);
    int m1=0,n1=0;
    pkg3 obj3=new pkg3();
    try
    {
```

```
    System.out.println("Enter First Number=");
    n1=Integer.parseInt(in.readLine());
    System.out.println("Enter Second Number=");
    m1=Integer.parseInt(in.readLine());
    obj3.x=m1;
    obj3.y=n1;
    obj3.product();
    }
    catch(Exception Sandhya){}

   }
}
```

Example:- 55

```
import java.io.*;
  class prg50
   {
   public static void main(String args[])
   {
     int a=12,b=0,c1;
   try
   {
    c1=a/b;
    System.out.println("Division="+c1);
   }
   catch(ArithmeticException m1)
   {
                    System.out.println("An error found Divide by Zero");

          }
   }
}
```

Mulitple catch Statements:-

Syntax:-

```
…
…
…
try
{
  Statement;
}
catch(Exception_Type_1 m1)
{
Statement1;
}
catch(Exception_Type_2 m2)
{
Statement2;
}
catch(Exception_Type_3 m3)
{
```

```
Statement3;
}
…
…
…
```

Using finally Statements:-

Syntax:-

```
…
…
…
try
{
  Statement;
}
catch(Exception_Type_1 m1)
{
Statement1;
}
catch(Exception_Type_2 m2)
{
Statement2;
}
catch(Exception_Type_3 m3)
{
Statement3;
}
finally
{
…
…
…
}
```

Example:- 56

```
import java.io.*;
  class prg51
    {
    public static void main(String args[])
    {
     DataInputStream in=new DataInputStream(System.in);
      int a1=12,b1=0,c1;
      int m[]=new int[4],i;
    try
    {
     c1=a1/b1;
      System.out.println("Division="+c1);
      System.out.println("Enter Elements in Single dimensional Array="+"\n");
      System.out.println("--------------------");
      for(i=0;i<=6;i++)
      m[i]=Integer.parseInt(in.readLine());
      System.out.println("--------------------");
      System.out.println("Enter Elements in Single dimensional Array="+"\n");
      for(i=0;i<=6;i++)
      System.out.print(m[i]+"\t");
    }
```

```
    catch(ArithmeticException m1)
    {
      System.out.println("An error found Divide by Zero b1");
    }
    catch(IOException m2)
    {
      System.out.println("An input output error");
    }
finally
{
System.out.println("Error can not be detected");
}
    }
}
```

LECTURE - 14

Overloading Method:-

The process of having two or more methods or functions within a class, with same name, but different parameter declaration is known as function overloading.

Example:-

```
import java.io.*;
class prg13ovr
{
public static void main(String args[])
{
BufferedReader in=new BufferedReader(new InputStreamReader(System.in));
float x=0,y=0;
int p=0,q=0;
 try
 {
 calculation obj1;
 obj1=new calculation();
System.out.println("Enter First float Number=");
x=Float.valueOf(in.readLine()).floatValue();
y=Float.valueOf(in.readLine()).floatValue();
obj1.f1(x,y);
System.out.println("Enter First integer Number=");
p=Integer.parseInt(in.readLine());
System.out.println("Enter second integer Number=");
q=Integer.parseInt(in.readLine());
obj1.f1(p,q);
}
catch(Exception akr_Rishi){}
}
}
class calculation
{
public static void f1(int p1,int q1)
{
int c1;
c1=p1+q1;
System.out.println("Sum of two number="+c1);
}
public static void f1(float x1,int q1)
{
float c3;
c3=x1-q1;
System.out.println("Difference of two number="+c3);
}
};
```

Java Overriding Method:-

In this technique of programming defining a method in the subclass that has the same name, same arguments and same return type as a method in the subclass.Then, when the method is called, the method defined in subclass is invoked and executed instead of one in the super class. This is known as overriding.

Or

Defining a function in the derived class with same name as in the parent class is called overriding. In java/C++, the base class member can be overridden by the derived class function with the same signature as the base class function. Method overriding is used to provide different implementations of a function so that a more specific behavior can be realized.

Characteristics of Overriding:-

1:- The return type should be the same or a subtype of the return type declared in the original overridden method in the super class.
2:- The access level cannot be more restrictive than the overridden method's access level.
3:- Instance methods can be overridden only if they are inherited by the subclass.
4:- A method declared final cannot be overridden.
5:- A method declared static cannot be overridden but can be re-declared.
6:- If a method cannot be inherited then it cannot be overridden.
7:- A subclass within the same package as the instance's super class can override any super class method that is not declared private or final.
8:- A subclass in a different package can only override the non-final methods declared public or protected.

Example:-

```
import java.io.*;
import java.util.Scanner;
class base
{
  public void show()
     {
     System.out.println("I am base");
  }
}
class derive extends base
  {
  public void show()
 {
     System.out.println("I am derive");
  }
}

public class prg50
{
  public static void main(String args[])
{
     base obj1 = new base();
     derive obj2 = new derive();
    obj1.show();
     obj2.show();
  }
}
```

Definition of constructor:-

Constructor in java is a special type of method that is used to initialize the object. Java constructor is invoked at the time of object creation. It constructs the values.

```
import java.io.*;
class base1
 {
 base1()
{
   System.out.println("Constructor method called.");
 }
};
class base2
 {
 base2()
{
   System.out.println("Constructor method called.");
```

```
 }
};
class prg51
{
  public static void main(String[] args)
{
    base1 object1 = new base1();
    base2 object2 = new base2();
  }
}
```

Question Bank of java

Question1:-What is OOPs? Explain with suitable example.

Question2:-What is Token in java? Explain with suitable example.

Question3:-What is datatypes? Describe with suitable example.

Question4:-What is role of bitwise operator? Describe with suitable example.

Question5:-What are different categories of control statement? Explain each.

Question6:- What is inheritance? Explain with suitable example.

Question7:- What is interface? Explain with suitable example.

Question8:- What is Applet? Explain with suitable example

Question9:- What is package? Explain with suitable example

Question10:- What is API? Explain with suitable example

Question11:- What is array? Explain with suitable example

Question12:- What is exception? How we handle exception in java.

Question13:- Write short notes of following.

- Java Exception class
- Where we use run( ) method
- sleep( ),wait( ) and start( )
- jdk

Question14:- Write java program of following

- Factorial
- fibonaccie
- Armstrong numbers
- Series of prime numbers
- Sorting of unordered numbers
- Matrix Addition
- Find Smallest and largest number in an array
- Area of triangle
- Area of circle and sphere
- Triangle of star
- Checking year is leap or not
- Solving of quadratic equation

Question15:-What is thread? How it is implemented in class

Question16:-Write a java program to draw a fill circle and fill rectangle.

Question17:-What is function? Describe each categories of functions.

Question18:-Write java expression of following

- $A=p(1+r/100)^n$
- $S=a^n+b^n$
- $Area=(4/3)3.14r^3$

Question19:-What will be output of following program

```
import java.io.*;
class pr3
{
 public static void main(String args[])
{
 int a=2,b=3;
 System.out.println("Value of a="+a+"\t"+"Value of b="+b);
a=++a + ++b;
  System.out.println("Value of a="+a+"\t"+"Value of b="+b);
 a=--a + b--;
System.out.println("Value of a="+a+"\t"+"Value of b="+b);
 }
}
```

Question20:-What will be output of following program

```
import java.io.*;
class pr3
{
 public static void main (String args[])
{
 int a=2, b=3, t1, t2, t3, t4, t5, t6, t7;
 t1=a&b;
 t2=a|b;
 t3=a^b;
 t4=~a;
 t5=a<<2;
 t6=b>>2;
 t7=a>>>2;
  System.out.println("Bitwise and="+t1);
  System.out.println("Bitwise OR="+t2);
  System.out.println("Bitwise Exclusive OR="+t3);
  System.out.println("Bitwise Compliment="+t4);
  System.out.println("Bitwise Left shift="+t5);
  System.out.println("Bitwise Right Shift="+t6);
  System.out.println("Bitwise Shift right With Zero="+t7);
 }
}
```