

20/7/24 UNIT-1 :- Introduction to OOPS :-

JAVA:-

JAVA is a high level & object oriented programming language. It was developed by "James Gosling" at "sun microsystems", USA. Actually, James gosling software initiating for electronic devices in 1991. That software named is "Oak". Oak is renamed as Hot Java, In 1994, But Jdk 1.1 version released in 1995.

oops (Object oriented programming) :-

- 1. Object
- 2. class
- 3. Inheritance
- 4. polymorphism
- 5. Data Abstraction
- 6. Encapsulation..

1. Object :-

Definition:- 1. An Instance of a class is called object. (or)
2. An object is a anything in the real world.

An object is Basic run time entities in an object oriented programming. An object consists 3 types of features, such as.

State :-

(It is represented by properties (or) Attributes (or))

Instance Variables.

Behaviour:-

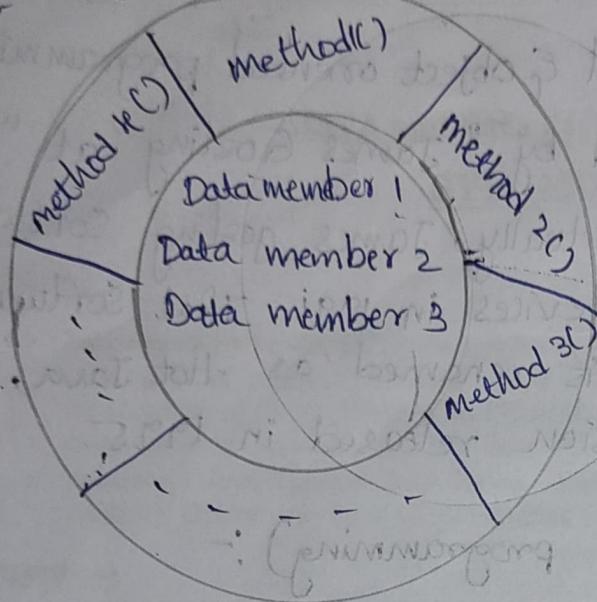
(It is represented by actions (or) methods (or) functional)

Identity:-

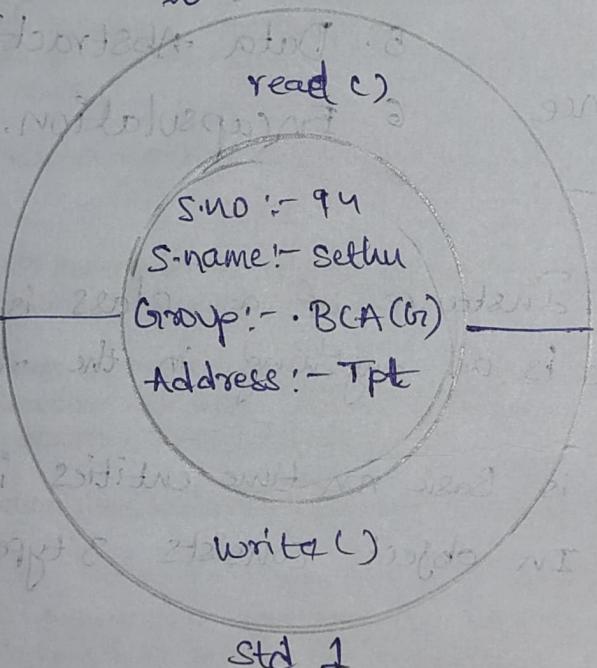
(It is unique name given to the created object.)

Actually object is a physical view of entity.

Ex:- Object Architecture.



Object :- Student Object :-



2) Class :-

Class is a user defined blueprint (or) prototype which objects are created. Actually class is a logical view of entity. Once class has been defined, we can create no. of class related objects. In that class consists 3 parts. such as,

- 1) class name
- 2) class Data member
- 3)

Class Architecture :-

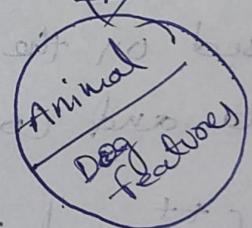
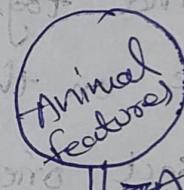
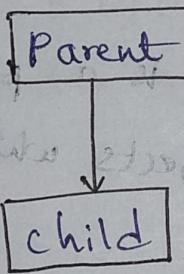
CLASS NAME
DATA MEMBER 1: TYPE
DATA MEMBER 2: TYPE
"
"
DATA MEMBER : TYPE
return type method()
return type method();

Ex:-

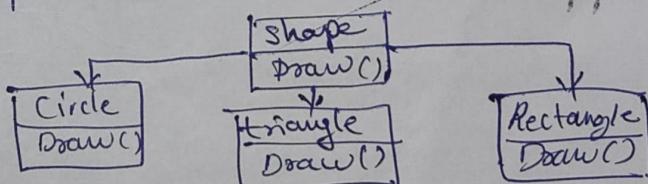
STUDENT
S.NO : 94
S.name : Sethu
Course : BCA
Void read()
Void write()

3) Inheritance: - It is a process of creating new class by reusing existing class.

The new class acquires all the properties and behaviours of an existing class. It is known as inheritance. It provides code reusability.



4) Polymorphism: If one task is performed in different ways it is known as polymorphism.



Hiding the internal details and showing functionality is known as Data Abstraction. (provide security)

For Ex:- phone call, we don't know the internal processing.

In Java class is a ADT (Abstract Data type)

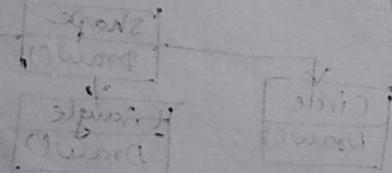
Q) Data Encapsulation:-

Binding code and data together into a single unit are known as Encapsulation.

Introduction to Object Oriented Programming :-

1. Object oriented programming refers to languages that use objects in program.
2. Object oriented programming aims to implement real world entities like Inheritance, hiding, polymorphism etc.
3. The main aim of object oriented programming is to bind together the data and the functions that operate on them.
4. Object oriented programming is a programming paradigm based on the concept of objects, which can contain data and code.

5. C++ and Java follows object oriented programming approach.



Tunability to take more than one form. This word belongs to greek.

Dynamic Binding

when type of object determined at runtime then it is called dynamic binding.

Write a program in Java to compute the following Segments.

1. area of rectangle & perimeter

2. Area of circumference & perimeter.

Class Area

{

float l, b, r, pi = 3.14f;

void Rectangle(float x, float y);

{

l = x;

b = y;

float ar = l * b;

float p = 2 * (l + b);

System.out.println("The Area of rectangle = " + ar);

System.out.println("The Perimeter = " + p);

}

void circle(float x)

{

r = x;

float ac = pi * r * r;

float p = 2 * pi * r;

```
System.out.println ("the area of circle = " + ac);
System.out.println ("the Perimeter = " + P);
}
public static void main (String [] args) {
    Area obj = new Area ();
    obj.Rectangle (3.4F, 5, 3F);
    Obj.Circle (5.5F);
}
```

② Biggest of two num using Scanner.

```
import java.util.Scanner;
```

```
Class Biggest
```

```
int a, b;
```

```
Scanner s = new Scanner (System.in);
```

```
void Scancompute ()
```

```
{ System.out.println ("Enter A, B values");
```

```
a = s.nextInt();
```

```
b = s.nextInt();
```

```
if (a > b)
```

```
{
```

```
System.out.println (a + " is the bigger value");
```

```
}
```

```
else if (a < b)
```

```
{
```

```
System.out.println (b + " is bigger value");
```

```
else
{
    System.out.println ("Both are equal");
}

}
public static void main (String []args)
{
    Biggest obj = new Biggest ();
    obj.compute ();
}
```

③ Sum of natural numbers up to a given number.

```
import java.util.Scanner;
class SNN {
    int n;
    Scanner s = new Scanner (System.in);
    void SNNsum()
    {
        System.out.println ("Enter a number:");
        n = s.nextInt();
        int sum=0;
        for (int i=1; i<=n; i++)
        {
            sum+=i;
        }
        System.out.println ("The sum of natural numbers = " + sum);
    }
    public static void main (String []args)
    {
```

SNN obj = new SNN();

obj = SNN sum();

}

Scope of Variables :-

① Static :-

Class scope

static int rno=0;
String name, course;
void set (String x, string y)

name = x;

class = y;

r.no = r.no+1;

System.out.println (" Roll number = " + rno);

System.out.println (" Student name = " + name);

System.out.println (" course = " + course);

}

public static void main (String [] args)

{

Scope obj = new Scope();

obj.set ("sethu", "BCA");

Scope obj2 = new Scope();

obj2.set ("varshini", "BSC");

{

② class Example

```
static int s=0; // * static variable *//  
int a,b; // * Instance variable *//  
void add (int x, int y) {  
    a=x;  
    b=y;
```

```
    int c=a+b; // * local variable *//
```

```
    s=s+1;
```

```
    System.out.println ("s=" + s + "sum=" + c);
```

```
}
```

```
public static void main (String args [])
```

```
{
```

```
    Example obj = new Example ();
```

```
    obj.add (10,20);
```

```
    Example obj2 = new Example ();
```

```
    obj2.add (10,50);
```

```
    Example obj3 = new Example ();
```

```
    obj3.add (50,50);
```

```
}
```

Scope of Java
variable

In Java variables are accessible only inside of the region. They are created. Then it is called
(Class or method)
Scope of variable.

The scope of variable refers to the portion of a program where the variable can be accessible. Scope means Area.

There are 3 types of scope of variables.

1. Instance variables

2. Local variables

3. Static variables (or) Class variables.

1. Instance Variable :-

An Instance variable are declared at inside of the class but not inside of the method. These variables are accessible inside of the ~~classes~~ class & methods.

2. Local Variables :-

The local variables are declared inside of the method. These variables are accessible only inside of the method.

3. Static Variable :-

Static Variable are declared inside of the class with key word using "static". These variables are accessible inside of the entire class & methods.

Static variables are create separate memory location but not create in object. static variables are also known as "Class Variables"

Type Conversion :-

" Type conversion is a process of converting one datatype of value to another datatype ". In type conversion, there are 2 types.

1. Implicit type conversion
2. Explicit type conversion.

Implicit type conversion :-

In this conversion, the lower data type values converting into higher datatype. This process is automatically done by "JVM", when assign a value of lower type to another type. This kind of conversion is also known as "NARROWING TYPE CONVERSION".

Ex:-

If you take short, float types

short a=10, float b= 8.5f

b=a
→
2 bytes %
 |
 |
 45
 |
 a

float type short of TC
→
4 bytes %
 |
 |
 10.500
 |
 b

- * Converts lower type value into higher type
- * this conversion automatically performed by

~~JVM~~ "JVM"

8.500

Programme

Class Implicit

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

```
{ Short a=45;  
float b;  
b=a;
```

```
System.out.println ("The implicit type conversion");
```

```
System.out.println ("a=" + a);
```

```
System.out.println ("b=" + b);
```

2. Explicit type Conversion

In this conversion the higher data type value converting into lower data type. This process manually done by programmer and by placing parenthesis "()" . It is also known as "widening type conversion" (o) "type casting".

Ex:-

If you take float, short

```
float a = 45.678F
```

Short b

b = (short) a

Type casting

4 bytes 45.678

a

2 bytes 45

b

* convert high type value into low type

* This conversion manually performed by user by placing "()"

Program:-

```

class Explicit
{
    public static void main (String [ ] args)
    {
        float a = 45.678F;
        short b;
        b = (short) a;
        System.out.println ("the explicit type conversion ");
        System.out.println (" a = " + a);
        System.out.println (" b = " + b);
    }
}

```

Programs:-

1. write a program in Java, compute factorial of given num.

```

import java.util.Scanner;
Class factorial

```

```

public static void main (String [ ] args)
{
    int n, fact = 1, a = 1;
}
```

```
Scanners = new Scanner (System.in);
```

```
System.out.println ("Enter n; value");
```

```
n = s.nextInt();
```

```
while (a <= n)
```

```

    fact = fact * a;
    a++;
}
```

```
System.out.println("The factorial of num = " + fact);
```

```
}
```

Q Write a program in java, to check whether the given num
is palindrome or not.

```
import java.util.Scanner;
```

```
class Palindrome.
```

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

```
{ int n, r, sum = 0, temp;
```

```
Scanner s = new Scanner (System.in);
```

```
System.out.println ("Enter number:");
```

```
n = s.nextInt();
```

```
temp = n;
```

```
while (n > 0)
```

```
{
```

```
    r = n % 10;
```

```
    sum = (sum * 10) + r;
```

```
    n = n / 10;
```

```
}
```

```
if (temp == sum)
```

```
{ System.out.println (temp + " ispalindrome num");
```

```
}
```

```
else
```

```
{
```

```
    System.out.println (temp + " isn't palindrome num");
```

```
}
```

```
}
```

```
}
```

③ write a programme in Java, to check whether the given num is Armstrong or not?

```

import java.util.Scanner;
class armstrong
{
    public static void main (String [] args)
    {
        int n, r, sum=0, temp;
        Scanner s = new Scanner (System. in);
        System.out.println ("Enter number:");
        n=s.nextInt();
        temp=n;
        while (n!=0)
        {
            r=n%10;
            sum= sum+r*r*r;
            n=n/10;
        }
        if (temp==sum)
        {
            System.out.println (temp + " is armstrong num");
        }
        else
        {
            System.out.println (temp + " isn't armstrong num");
        }
    }
}

```

④ write a programme in Java, to print the Fibonacci Series.

```
import java.util.Scanner;
```

```
class Fibonacci
```

```
{
```

```
    public static void main (String args [])
```

```
{
```

```
        int n, a=0, b=1, c;
```

```
        Scanner s = new Scanner (System.in);
```

```
        System.out.println ("Enter num: ");
```

```
        n=s.nextInt();
```

```
        System.out.println ("The Fibonacci series are ");
```

```
        System.out.print (a + " " + b);
```

```
        while (n != 0)
```

```
{
```

```
    c=a+b;
```

```
    System.out.print (" " + c);
```

```
    a=b;
```

```
    b=c;
```

```
    n--;
```

```
}
```

```
}
```

```
(Even problem 2) (Even problem 2)
```