\*oops in java

Q)What is oops?

Object-Oriented Programming or OOPs refers to languages that use objects in programming, they use objects as a primary source to implement what is to happen in the code. Objects are seen by the viewer or user, performing tasks assigned by you. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc. in programming

Acces Modifier:-

public: Accessible in all classes in your application.

protected: Accessible within the package in which it is defined and in its subclass(es) (including subclasses declared outside the package).

private: Accessible only within the class in which it is defined.

default (declared/defined without using any modifier): Accessible within the same class and package within which its class is defined.

The return type: The data type of the value returned by the method or void if it does not return a value.

Method Name: The rules for field names apply to method names as well, but the convention is a little different.

Parameter list: Comma-separated list of the input parameters that are defined, preceded by their data type, within the enclosed parentheses. If there are no parameters, you must use empty parentheses ().

Exception list: The exceptions you expect the method to throw. You can specify these exception(s).

Method body: It is the block of code, enclosed between braces, that you need to execute to perform your intended operations.

Message passing:-

1.Class

2.Object

3.Method and method passing

4.Pillars of OOPs

Abstraction

Encapsulation

Inheritance

Polymorphism

Compile-time polymorphism

Runtime polymorphism

A)Class:-

A class in java can contain:

• data member

• method

• constructor

• nested class and

• interface

Types of class:-

1.nested classes

2.anonymous classes

3.lambda expressions.

Syntax to declare a class:

access\_modifier class<class\_name>

{

data member;

method;

constructor;

nested class;

interface;

}

Code:-

package CLAss;

public class assigne {

    String name="prince";

    int rollno=7;

    char sec='B';

    public static void main(String argrs[]) {

        assigne p1=new assigne();

        System.out.println(p1.name);

        System.out.println(p1.rollno);

        System.out.println(p1.sec);

    }

}

Output:-

5

Prince

1.Nested Class:-

it is possible to define a class within another class, such classes are known as nested classes.

Syntax:

Code:-

package CLAss;

class Outerclass{

    static int outer\_x=10;

    int outer\_y=20;

    private int outer\_private=30;

    class Innerclass{

        void display(){

            System.out.println("outer\_x=" + outer\_x);

            System.out.println("outer\_y" + outer\_y);

            System.out.println("outer\_private" + outer\_private);

        }

    }

}

public class nest\_CLass {

    public static void main(String args[]) {

        // accessing an inner class

        Outerclass outerobj = new Outerclass();

        Outerclass.Innerclass innerobj = outerobj.new Innerclass();

        innerobj.display();// calling a constructor

    }

}