**Name : Muhammad Murad**

**Roll no : 21k3064**

**Inner Class**

Inner classes in Java are the classes that are defined inside the scope of another class. Inner classes are also called nested classes. This helps in easy documentation and better maintenance of the code.

**Types**

1. Local inner class
2. Member inner class
3. Static Member inner class
4. Anonymous class

**Local inner class**

A class created inside a method is called a local inner class in java. Local Inner Classes are the inner classes that are defined inside a block. Generally, this block is a method body.

public class localInner1{

private int data=2;

void display(){

class Local{

void msg(){System.out.println(data);}

}

Local l=new Local();

l.msg();

}

public static void main(String args[]){

localInner1 obj=new localInner1();

obj.display();

}

}

**Usage**

Whenever we want to use a particular class at a particular moment only, like whenever we want to use it basically supports encapsulation.

**Member inner classes**

A non-static class that is created inside a class but outside a method is called member inner class. It is also known as a regular inner class. It can be declared with access modifiers like public, default, private, and protected.

class MemberOuter{

private int data=5;

class Inner{

void msg(){System.out.println("data is "+data);}

}

public static void main(String args[]){

MemberOuter obj=new MemberOuter();

MemberOuter.Inner in=obj.new Inner();

in.msg();

}

}

**Usage**

To access the outer class private data within the inner class. Whenever we want to encapsulate something and at that time we want to use that data.

**Static Member Inner Class**

A static class is a class that is created inside a class, is called a static nested class in Java. It cannot access non-static data members and methods. It can be accessed by outer class name. It can access static data members of the outer class, including private.

public class OuterClass {

int outerVariable = 100;

static int staticOuterVariable = 200;

static class StaticMemberClass {

int innerVariable = 20;

int getSum(int parameter) {

// Cannot access outerVariable here.

return innerVariable + staticOuterVariable + parameter;

}

}

public static void main(String[] args) {

OuterClass outer = new OuterClass();

StaticMemberClass inner = new StaticMemberClass();

System.out.println(inner.getSum(3));

outer.run();

}

void run() {

StaticMemberClass localInner = new StaticMemberClass();

System.out.println(localInner.getSum(5));

}

}

**Usage**

In static nested class object of inner class don't need object of outer class, because the word "static" indicate no need to create object. Show activity on this post. The instance of the inner class is created when instance of the outer class is created.

**Anonymous Class**

Java anonymous inner class is an inner class without a name and for which only a single object is created. An anonymous inner class can be useful when making an instance of an object with certain "extras" such as overloading methods of a class or interface, without having to actually subclass a class.

abstract class Person{

abstract void run();

}

class TestAnonymousInner{

public static void main(String args[]){

Person p=new Person(){

void run(){System.out.println("running");}

};

p.run();

}

}

Usage

Used in GUI most , action event listeners , buttons.

**Question 2 :**

Read and write into binary files

WRITE

FileOutputStream fos = new (“abc.txt”);

String a = “hello”;

byte[] b = a.getbytes();

fos.write(b);

fos.close;

READ

FileInputStream fis = new (“abc.txt”);

int r = 0;

while((r = fis.read()) != -1)

{

sout ((char)1);

}

Read and write into txt files

**Writing**

FileWrite fw = new FileWriter(“hello.txt”);

fw.write(“hello”);

fw.close();

**Read**

FileReader fr=new FileReader("D:\\testout.txt");

int i;

while((i=fr.read())!=-1)

System.out.print((char)i);

fr.close();

Or we can also read by Scanner