**AbstractDemo.java:-**

// understanding the abstarct classes and abstract methods

/\*\*

\* **@author** Lakshman

\*

\*/

**abstract** **class** c1

{

**abstract** **void** f1();

**abstract** **void** f2();

}//c1

**abstract** **class** c2 **extends** c1

{

**void** f1()

{

System.*out*.println("f1() method defined in class c2");

}

}//c2

**class** c3 **extends** c2

{

**void** f2()

{

System.*out*.println("f2() method defined in class c3");

}

**void** f1()

{

//super.f1();

System.*out*.println("f1() method re-defined in class c3");

}

**void** f3()

{

System.*out*.println("f3() method belongs to class c3");

}

}//c3

**class** AbstractDemo

{

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

{

System.*out*.println("\nwith respect to class c3 object\n");

c3 o3=**new** c3();

o3.f1();

o3.f2();

o3.f3();

System.*out*.println("\nwith respect to class c2 object\n");

//c2 o2=new c2(); invalid, since c2 is abstract

c2 o2;

o2=**new** c3();

o2.f1();

o2.f2();

//o2.f3(); invalid, since f3() does not exists in c2

System.*out*.println("\nwith respect to class c1 object\n");

//c1 o1=new c1();//Invalid

c1 o1=**new** c3();

o1.f1();

o1.f2();

//o1.f3(); invalid, since f3() does not exists in c1

}

}

**AbstactCls.java:-**

//abstract method is a method it doesn't contain body.

//abstract class is a class which does not contain body.

/\*\*

\* **@author** Lakshman

\*

\*/

**abstract** **class** Myclass2{

**abstract** **void** cal(**double** x);

}

**class** ASub1 **extends** Myclass2{

**void** cal(**double** x){

System.*out*.println("square of "+(x\*x));

}

}

**class** ASub2 **extends** Myclass2{

**void** cal(**double** x){

System.*out*.println("cube of "+(x\*x\*x));

}

}

**class** ASub3 **extends** Myclass2{

**void** cal(**double** x){

System.*out*.println("square of "+(x\*x\*x\*x));

}

}

**public** **class** AbstactCls {

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

ASub1 a1=**new** ASub1();

ASub2 a2=**new** ASub2();

ASub3 a3=**new** ASub3();

a1.cal(2);

a2.cal(3);

a3.cal(4);

}

}

**AbstrDemo.java:-**

/\*\*

\* **@author** Lakshman

\*

\*/

**abstract** **class** Operation

{

**abstract** **void** sum();

}

**class** isum **extends** Operation

{

**int** a,b,c;

**void** sum()

{

a=1;

b=2;

c=a+b;

System.*out*.println(c);

}

}

**class** fsum **extends** Operation

{

**float** f1,f2,f3;

**void** sum()

{

f1=1.1f;

f2=2.2f;

f3=f1+f2;

System.*out*.println(f3);

}

}

**class** AbstrDemo

{

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

{

Operation op;

op=**new** isum();

op.sum();

Operation op1;

op1=**new** fsum();

op1.sum(); }

}

Procedure:- right clink on package->New->interface(click)

**InterfaceDemo.java:-**

/\*\*

\* **@author** Lakshman

\*

\*/

**interface** i1

{

**public** **void** f1();

}

**interface** i2 **extends** i1

{

**public** **void** f2();

}

**class** cls1

{

**void** f3()

{

System.*out*.println("f3 defined in c1");

}

}

**abstract** **class** cls2 **extends** cls1 **implements** i2

{

**public** **void** f1()

{

System.*out*.println("f1 defined in c2");

}

**public** **void** f3()

{

**super**.f3();

System.*out*.println("f3 redefined in c2");

}

}

**class** cls3 **extends** cls2

{

**public** **void** f2()

{

System.*out*.println("f2 defined in c3");

}

**public** **void** f1()

{

**super**.f1();

System.*out*.println("f1 defined in c3");

}

**public** **void** f3()

{

System.*out*.println("f3 redefined in c3");

}

**public** **void** f4()

{

System.*out*.println("f4 defined in c3");

}

}

**class** InterfaceDemo

{

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

{

cls3 o3=**new** cls3();

o3.f1(); //f1 defined in c2,f1 defined in c3

o3.f2();//f2 defined in c3

o3.f3();//f3 redefined in c3

o3.f4();//f4 defined in c3

cls2 o2=**new** cls3();

o2.f1();//f1 defined in c2,f1 defined in c3

o2.f2();//f2 defined in c3

o2.f3();//f3 redefined in c3

cls1 o1=**new** cls3();

o1.f3();//f3 redefined in c3

i2 ioi=**new** cls3();

ioi.f1();//f1 defined in c2,f1 defined in c3

ioi.f2();//f2 defined in c3

i1 io=**new** cls3();

io.f1();//f1 defined in c2,f1 defined in c3 }

}

**MyInter.java:-**

/\*\*

\* **@author** Lakshman

\*

\*/

**public** **interface** MyInter {

**public** **void** connect();

**public** **void** disconnect();

}

===========================

**MyInterF.java:-**

/\*\*

\* **@author** Lakshman

\*

\*/

**public** **interface** MyInterF {

**abstract** **void** method1();

}

============================

**InterfaceDemo1.java:-**

//an interface is a specification of method prototype.All the methods of

//interface are public and abstract

/\*\*

\* **@author** Lakshman

\*

\*/

**public** **class** InterfaceDemo1 **implements** MyInter,MyInterF{

**public** **void** method1(){

System.*out*.println("hi this is multi-inheritance");

}

**public** **void** connect(){

System.*out*.println("connect");

}

**public** **void** disconnect(){

System.*out*.println("disconnect");

}

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

InterfaceDemo1 i=**new** InterfaceDemo1();

i.method1();

i.connect();

i.disconnect();

}

}

**InterfaceStackTest.java:-**

/\*\*

\* **@author** Lakshman

\*

\*/

**interface** IntStack

{

**void** push(**int** item);

**int** pop();

}

**class** FixedStack **implements** IntStack

{

**private** **int** stck[];

**private** **int** tos;

FixedStack(**int** size)

{

stck=**new** **int**[size];

tos=-1;

}

**public** **void** push(**int** item)

{

**if**(tos==stck.length-1)

System.*out*.println("Stack is full");

**else**

stck[++tos]=item;

}

**public** **int** pop()

{

**if**(tos<0)

{

System.*out*.println("Stack underflow");

**return** 0;

}

**else**

**return** stck[tos--];

}

}

**class** InterfaceStackTest

{

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

{

IntStack ob;

FixedStack mystck1=**new** FixedStack(8);

FixedStack mystck2=**new** FixedStack(8);

ob=mystck1;

**for**(**int** i=0;i<8;i++)

mystck1.push(i);

ob=mystck2;

**for**(**int** i=0;i<8;i++)

mystck2.push(i);

ob=mystck1;

System.*out*.println("Stack in mystack1:");

**for**(**int** i=0;i<8;i++)

System.*out*.println(mystck1.pop());

ob=mystck2;

System.*out*.println("Stack in mystack2:");

**for**(**int** i=0;i<8;i++)

System.*out*.println(mystck2.pop());

}

}

Procedure:- right clink on src->New->package(click)-> package name:pack

**Addition.java:-**

//creating package pack with addition class

**package** pack;

/\*\*

\* **@author** Lakshman

\*

\*/

**public** **class** Addition {

**private** **double** d1,d2;

**public** Addition(**double** a,**double** b){

d1=a;

d2=b;

}

**public** **void** sum(){

System.*out*.println("sum of : "+(d1+d2));

}

}

===================

UsePack.java:-

**package** pack;

/\*\*

\* **@author** Lakshman

\*

\*/

**public** **class** UsePack {

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

//create addition class object

pack.Addition o=**new** pack.Addition(10,15.6);

//call the sum() method

o.sum();

}

}

Procedure:- right clink on src->New->package(click)-> package name:mypack

**Sum.java:-**

// PACKAGE APPLICATION

/\*\*

\* **@author** Lakshman

\*

\*/

**package** mypack; // Must be first executable statement

**public** **class** sum

{

**int** a,b,c;

**public** **void** assign(**int** x,**int** y)

{

a=x;

b=y;

}

**public** **void** add() // public is mandatory

{

c=a+b;

}

**public** **void** show() // public is mandatory

{

System.*out*.println("The value of a : "+a);

System.*out*.println("The value of b : "+b);

System.*out*.println("The sum is : "+c);

}

**public** sum()

{

System.*out*.println("\nZero argument constructor of sum class");

}

}//class

========================

Procedure:- right clink on src->New->package(click)-> package name:pack1

**SumDemo.java:-**

**package** pack1;

/\*\*

\* **@author** Lakshman

\*

\*/

// using the class that is there in mypack package

//import mypack.\*;

**import** mypack.sum;

**class** SumDemo

{

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

{

//mypack.sum s=new mypack.sum();//F.Q.N.A

sum s=**new** sum();

s.assign(100,200);

s.add();

s.show();

}//main()

}//class