1. **Write a Java Program to demonstrate the concept of method overriding.**

class metride

{

public void print()

{

System.out.println("From base class");

}

}

class metride1 extends metride

{

public void print()

{

System.out.println("From derived class");

}

public static void main(String args[])

{

metride1 m=new metride1();

System.out.println("Example of Method Overriding");

m.print();

}

}

**11.Write a JAVA Program to implement Inner class and demonstrate its Access Protections.**

class outer

{

int a=10;

class inner

{

int b=20;

}

}

class test

{

public static void main(String args[])

{

outer o=new outer();

outer.inner i=o.new inner();

System.out.println("Example of Inner Class access protection");

System.out.println(i.b+o.a);

}

}

**12.Write a Java program to demonstrate wrapper class.**

class wrap

{

public static void main(String args[])

{

int a=5;

double d=10.5;

Integer aobj=Integer.valueOf(a);

Double bobj=Double.valueOf(d);

if(aobj instanceof Integer)

{

System.out.println("Interger object is created");

}

if(bobj instanceof Double)

{

System.out.println("Double object created");

}

}

}

**13.Write a Java Program to demonstrate single inheritance.**

class single {

void display()

{

System.out.println("Inside Base Class");

}

}

class single1 extends single{

void display1()

{

System.out.println("Inside Derived Class");

}

public static void main(String args[]){

single1 s=new single1();

System.out.println("Example of single inheritance");

s.display();

s.display1();

}

}

}

**14.Write a Java program to demonstrate concept of multilevel inheritance.**

class A {

void displayA()

{

System.out.println("Inside Base Class A");

}

}

class B extends A{

void displayB()

{

System.out.println("Inside Derived Class B");

}

class C extends B{

void displayC()

{

System.out.println("Inside Derived Class C");

}

public static void main(String args[]){

C obj=new C();

System.out.println("Example of multilevel inheritance");

obj.displayA();

obj.displayB();

obj.displayC();

}

}

}

**15.Write a Java Program to demonstrate the working of Super keyword.**

class student

{

String college="RLS BCA";

}

class student1 extends student{

String college="KLE BCA";

void printcollege(){

System.out.println(“College Name is”+college);//prints college of student class

System.out.println(“College Name is”+super.college);//prints college of student1 class

}

public static void main(String args[])

{

student1 s=new student1();

System.out.println(“Example of super keyword”);

s.printcollege();

}

}

**16.Write a Java Program to demonstrate the concept of Interface.**

interface inter

{

public void print();

}

class inter1 implements inter

{

public void print()

{

System.out.println("Example of interface");

}

public static void main(String args[])

{

inter1 i=new inter1();

i.print();

}

}

**17.Write a Java program to demonstrate working of multiple inheritance using interface.**

interface A

{

public void print();

}

interface B

{

public void print();

}

class mulin implements A,B

{

public void print()

{

System.out.println("From First Method");

}

public static void main(String args[])

{

mulin m=new mulin();

System.out.println(“Example of multiple inheritance using interface”);

m.print();

}

}

**18.Write a Java Program to demonstrate the concept of Abstract class and Abstract methods.**

abstract class abs

{

abs()

{

System.out.println("Constructoe in abstract class");

}

void normal()

{

System.out.println("Normal Method in abstract class");

}

abstract void print();

}

class abst extends abs

{

void print()

{

System.out.println("Example of abstract class and method");

}

public static void main(String args[])

{

abst a=new abst();

a.print();

a.normal();

}

}

**19.Write a Java Program to demonstrate the concept of static keyword with respect to memory management.**

class stat

{

int id;

String name;

static String college="RLS";

stat(int i,String n)

{

id=i;

name=n;

}

public void print()

{

System.out.println(id+" "+name+" "+college);

}

}

class stat1

{

public static void main(String args[])

{

stat s=new stat(111,"Rohit");

stat s2=new stat(222,"ABC");

System.out.println(“Example of static keyword”);

s.print();

s2.print();

}

}

**20. Write a Java program to demonstrate user defined packages in Java.**

A.java

package pack;

public class A

{

public void print()

{

System.out.println("Example of user defined packages");

}

}

B.java

import pack. A;

public class B

{

public static void main(String args[])

{

A obj=new A();

obj.print();

}

}

**21.Write a Java Program to demonstrate the working of try, catch and finally block in exception handling.**

class ex

{

public static void main(String args[])

{

try

{

int data=100/0;

System.out.println(data);

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

System.out.println("Example of try, catch and finally");

}

}

}

**22.Write a Java Program to demonstrate the working of multiple catch blocks.**

class ex

{

public static void main(String args[])

{

try

{

int []a={10,20,30,40,50};

System.out.println(a[6]);

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("Array index exception occured");

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

System.out.println("Example of multiple catch");

}

}

}

**23.Write a Java Program to demonstrate the working of Nested try block.**

class nestedtry

{

public static void main(String args[])

{

try

{

try

{

int data=100/0;

}

catch(Exception e)

{

System.out.println(e);

}

try

{

int[] a={1,2,3,4,5};

System.out.println(a[10]);

}

catch(Exception e)

{

System.out.println(e);

}

}

finally

{

System.out.println("Example of nested try");

}

}

}

**24.Write a Java program to demonstrate the working of Throw keyword.**

class thr

{

public static void validate(int age)

{

if(age<18)

{

throw new ArithmeticException("Not eligible");

}

else

{

System.out.println("Eligible");

}

}

public static void main(String args[])

{

validate(22);

System.out.println("Example of throw");

}

}