01.write a program that demonstrate program for factorial number

import java.util.Scanner;

public class fac

{

public static void main(String[] args)

{

int n, i, fact=1;

Scanner s = new Scanner(System.in);

System.out.print("Enter a Number: ");

n = s.nextInt();

for(i=n; i>=1; i--)

{

fact = fact\*i;

}

System.out.println("\nFactorial Result = " +fact);

}

}

02.write a java program to demonstrate use of class and object.

class Student{

int rollno;

String name;

}

class TestStudent{

public static void main(String args[]){

Student s1=new Student();

s1.rollno=01;

s1.name="raj";

System.out.println(s1.rollno+" "+s1.name);//printing members with a white space

}

}

03. write a java program that demonstrate all string operations.

class string

{

public static void main(String[] args)

{

// create a string

String first = "Hello";

String second = "world";

String third = "world";

System.out.println("first: " + first);

System.out.println("Second: " + second);

// join two strings

String join = first.concat(second);

System.out.println("Join String: " + join);

// get the length of greet

int length = join.length();

System.out.println("Length: " + length);

// compare first and second strings

boolean result1 = first.equals(second);

System.out.println("Strings first and second are equal: " + result1);

// compare second and third strings

boolean result2 = second.equals(third);

System.out.println("Strings second and third are equal: " + result2);

}

}

04.write a java program to demonstrate use of constructor and finalize method.

class con {

// member variable

int a;

// constructor

con(int a) {

System.out.println("Inside the con class constructor.");

this.a = a;

System.out.println("Awesomeness set!");

}

// method

public int getA() {

return this.a;

}

// finalize

protected void finalize() {

System.out.println("Inside the con class finalize method...");

System.out.println("Object getting destroyed... Bye...");

}

}

class AwesomeExample {

public static void main(String[] args) {

con obj = new con(10);

System.out.println("Awesomeness of the object is " + obj.getA());

}

}

05.write a java program to demonstrate use of method overloading.

class MethodOverloading

{

void display(int a)

{

System.out.println("Arguments: " + a);

}

void display(int a, int b)

{

System.out.println("Arguments: " + a + " and " + b);

}

public static void main(String[] args)

{

MethodOverloading m1=new MethodOverloading();

m1.display(1);

m1.display(1, 4);

}

}

06. write a java program to demonstrate use of wrapper class

import java.lang.\*;

public class Wrapper

{

public static void main(String args[])

{

int a=20;

Integer i=Integer.valueOf(a);

Integer j=a;

System.out.println(a+" "+i+" "+j);

}

}

07. write a java program to demonstrate use of package.

//save by A.java

**package** pack;

**public** **class** A{

**public** **void** msg(){System.out.println("Hello");}

}

//save by B.java

**package** mypack;

**import** pack.\*;

**class** B{

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

  A obj = **new** A();

   obj.msg();

  }

}

08.write a java program that demonstrate inheritance.

class A

{

public void methodA()

{

System.out.println("Base class method");

}

}

class B extends A

{

public void methodB()

{

System.out.println("Child class method");

}

}

class demo

{

public static void main(String args[])

{

B obj = new B();

obj.methodA();

obj.methodB();

}

}

09. write a java program to demonstrate interface.

interface Printable{

void print();

}

interface Showable{

void show();

}

class test

{

void hello()

{

System.out.println("test class");

}

}

class inter extends test implements Printable, Showable

{

public void print()

{

System.out.println("Hello");

}

public void show()

{

System.out.println("Welcome");

}

public static void main(String args[])

{

inter obj = new inter();

obj.print();

obj.show();

}

}

10. write a java program that demonstrates inner class

class Outer {

// inner class

private class Inner {

public void print() {

System.out.println("This is an inner class");

}

}

// Accessing he inner class from the method within

void displayInner() {

Inner inner = new Inner();

inner.print();

}

}

public class Myclass {

public static void main(String args[]) {

// Instantiating the outer class

Outer outer = new Outer();

// Accessing the displayInner() method.

outer.displayInner();

}

}

11. write a java program that demonstrates exception (divide by 0).

import java.io.\*;

class zero {

public static void main(String[] args)

{

int a = 5;

int b = 0;

try {

System.out.println(a / b); // throw Exception

}

catch (ArithmeticException e) {

// Exception handler

System.out.println(

"Divided by zero operation cannot possible");

}

}

}

12.write a java program that demonstrates AWT control (label, textbox,button,etc).

import java.awt.\*;

import java.awt.event.\*;

// creating class which implements ActionListener interface and inherits Frame class

public class LabelExample2 extends Frame implements ActionListener{

// creating objects of TextField, Label and Button class

TextField tf;

Label l;

Button b;

// constructor to instantiate the above objects

LabelExample2() {

tf = new TextField();

tf.setBounds(50, 50, 150, 20);

l = new Label();

l.setBounds(50, 100, 250, 20);

b = new Button("Find IP");

b.setBounds(50,150,60,30);

b.addActionListener(this);

add(b);

add(tf);

add(l);

setSize(400,400);

setLayout(null);

setVisible(true);

}

// defining actionPerformed method to generate an event

public void actionPerformed(ActionEvent e) {

try {

String host = tf.getText();

String ip = java.net.InetAddress.getByName(host).getHostAddress();

l.setText("IP of "+host+" is: "+ip);

}

catch (Exception ex) {

System.out.println(ex);

}

}

// main method

public static void main(String[] args) {

new LabelExample2();

}

}

13.write a java program to print Fibonacci series.

class Fibonacci {

public static void main(String[] args) {

int n = 100, first = 0, second = 1;

System.out.println("Fibonacci Series Upto " + n + ": ");

while (first <= n) {

System.out.print(first + ", ");

int next = first + second;

first = second;

second = next;

}

}

}

14. write a java program to check given number is armstrong or not.

import java.util.Scanner;

public class Armstrong {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int number, originalNumber, remainder, result = 0;

System.out.println("enter the number");

number=sc.nextInt();

originalNumber = number;

while (originalNumber != 0)

{

remainder = originalNumber % 10;

result += Math.pow(remainder, 3);

originalNumber /= 10;

}

if(result == number)

System.out.println(number + " is an Armstrong number.");

else

System.out.println(number + " is not an Armstrong number.");

}

}

15. write a java program to check given number is palindrome or not.

import java.util.Scanner;

class pal {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int num, reversedNum = 0, remainder;

System.out.println("enter the number");

num=sc.nextInt();

int originalNum = num;

while (num != 0) {

remainder = num % 10;

reversedNum = reversedNum \* 10 + remainder;

num /= 10;

}

if (originalNum == reversedNum) {

System.out.println(originalNum + " is Palindrome.");

}

else {

System.out.println(originalNum + " is not Palindrome.");

}

}

}

16. write a java program to check given number is perfect number or not.

import java.util.Scanner;

public class Perfect

{

public static void main(String args[])

{

long n,

sum=0;

Scanner sc=new Scanner(System.in);

System.out.print("Enter the number: ");

n=sc.nextLong();

int i=1;

while(i <= n/2)

{

if(n % i == 0)

{

sum = sum + i;

}

i++;

}

if(sum==n)

{

System.out.println(n+" is a perfect number.");

}

else

System.out.println(n+" is not a perfect number.");

}

}

17. write a java program to print addition of digits.

import java.util.Scanner;

public class add

{

public static void main(String args[])

{

int number, digit, sum = 0;

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number: ");

number = sc.nextInt();

while(number > 0)

{

//finds the last digit of the given number

digit = number % 10;

//adds last digit to the variable sum

sum = sum + digit;

//removes the last digit from the number

number = number / 10;

}

//prints the result

System.out.println("Sum of Digits: "+sum);

}

}

18. write a java program to demonstrate use of method overriding

class Parent {

void show()

{

System.out.println("Parent's show()");

}

}

class Child extends Parent {

@Override

void show()

{

System.out.println("Child's show()");

}

}

class Overriding{

public static void main(String[] args)

{

Parent obj1 = new Parent();

obj1.show();

Parent obj2 = new Child();

obj2.show();

}

}

19. write a java program to demonstrate use of final class.

final class finalclass {

public void display() {

System.out.println("This is a final method.");

}

}

class Main extends finalclass {

public void display() {

System.out.println("The final method is overridden.");

}

public static void main(String[] args) {

Main obj=new Main();

obj.display();

}

}

21. write a java program to demonstrate abstract.

abstract class X

{

int i = 10;

public abstract void display(); //abstract method

}

class Y extends X

{

int i = 20;

public void display()

{

System.out.println("hello");

}

}

class abs

{

public static void main(String args[])

{

Y d = new Y();

d.display();

System.out.println(" d.i" + d.i);

X a = new Y(); //reference of abstract class

a.display();

System.out.println(" a.i " + a.i);

}

}

22. write a java program to demonstrate applet programming.

import java.applet.Applet;

import java.awt.Graphics;

public class HelloWorld extends Applet

{

public void paint(Graphics g)

{

g.drawString("Hello World", 20, 20);

}

}

23. write an applet program to display a various shape in applet.

**import** java.applet.Applet;

**import** java.awt.\*;

**public** **class** GraphicsDemo **extends** Applet{

**public** **void** paint(Graphics g){

g.drawString("Welcome",50, 50);

g.drawLine(20,30,20,300);

g.drawRect(70,100,30,30);

g.fillRect(170,100,30,30);

g.drawOval(70,200,30,30);

g.fillOval(170,200,30,30);

g.drawArc(90,150,30,30,30,270);

g.fillArc(270,150,30,30,0,180);

}

}

myapplet.html

<html>

<body>

<applet code="GraphicsDemo.class" width="300" height="300">

</applet>

</body>

</html>

24. write an applet program to display human face.

import java.applet.\*;

import java.awt.\*;

public class Smiley extends Applet {

public void paint(Graphics g)

{

// Oval for face outline

g.drawOval(80, 70, 150, 150);

// Ovals for eyes

// with black color filled

g.setColor(Color.BLACK);

g.fillOval(120, 120, 15, 15);

g.fillOval(170, 120, 15, 15);

// Arc for the smile

g.drawArc(130, 180, 50, 20, 180, 180);

}

}

Smiley.html

<html>

<body>

<applet code ="Smiley" width=600 height=600>

</applet>

</body>

</html>

25. write a java program that demonstrates AWT control (addition of two number).

import java.awt.\*;

import java.net.\*;

import java.awt.event.\*;

public class LearnAWT extends Frame {

TextField tf1;

TextField tf2;

Label l1;

Button b;

LearnAWT() {

setTitle("Adder");

tf1 = new TextField();

tf1.setBounds(100, 50, 85, 20);

tf2 = new TextField();

tf2.setBounds(100, 100, 85, 20);

b = new Button("Add");

b.setBounds(110,220,60,40);

l1 = new Label("");

l1.setBounds(100, 120, 85, 20);

add(b);

add(tf1);

add(tf2);

add(l1);

setSize(300,300);

setVisible(true);

b.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e) {

int a = Integer.parseInt(tf1.getText());

int b = Integer.parseInt(tf2.getText());

int c = a + b;

l1.setText("Their sum is = " + String.valueOf(c));

}

});

}

public static void main(String []args) {

new LearnAWT();

}

}

26. write an applet program to display Indian flag.

import java.applet.Applet;

import java.awt.\*;

public class flag extends Applet

{

public void paint(Graphics g)

{

g.drawRect(20,20,10,300);//pole

g.setColor(Color.orange);//upper square

g.fillRect(30,20,90,20);

g.setColor(Color.white);//centre square

g.fillRect(30,40,90,20);

g.setColor(Color.green);//lower square

g.fillRect(30,60,90,20);

g.setColor(Color.black);//upper square border

g.drawRect(30,20,90,20);

g.setColor(Color.black);//centre square border

g.drawRect(30,40,90,20);

g.setColor(Color.black);//lower square border

g.drawRect(30,60,90,20);

g.setColor(Color.blue);//centre circle

g.drawOval(68,43,14,14);

}

}