## Program1: implement of data types and variable

public class HelloWorld{

public static void main(String []args){

byte num1=100;

short num2=5000;

int num3=10000;

long num4=1200000l;

float num5=6.75f;

double num6=19.79d;

boolean flag1=true;

boolean flag2=false;

char grade='A';

String str="hello world";

System.out.println("byte:"+num1);

System.out.println("short:"+num2);

System.out.println("integer:"+num3);

System.out.println("long:"+num4);

System.out.println("float:"+num5);

System.out.println("double:"+num6);

System.out.println("boolean value 1:"+flag1);

System.out.println("boolean value 2:"+flag2);

System.out.println("character:"+grade);

System.out.println("string:"+str);

int a=10;

float f=a;

System.out.println(a);

System.out.println("after widening:"+f);

float b=10.5f;

int c=(int)b;

System.out.println(b);

System.out .println("after narrowing:"+c);

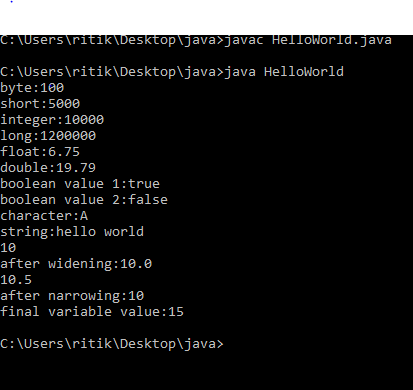
final int mynum=15;

System.out.println("final variable value:"+mynum)

}

}

## Output:



## Program 2: parametrized constructor

class Student{

int id;

String name;

int age;

Student(int i,String n){

id=i;

name=n;

}

Student (int i,String n,int a){

id=i;

name=n;

age=a;

}

void display(){

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

}

public static void main(String args[]){

Student s1=new Student (100,"aman");

Student s2=new Student(200,"rakesh",21);

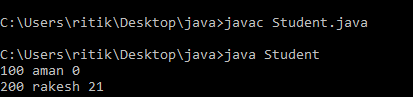
s1.display();

s2.display();

}

}

### Output:



## Program 3:

class Studenty {

int id;

String name;

}

class TestStudent{

public static void main(String args[]){

Studenty s1 =new Studenty();

Studenty s2 =new Studenty();

s1.id=101;

s2.id=201;

s1.name="anil";

s2.name="vishal";

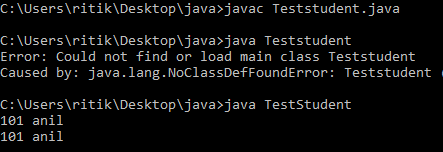
System.out.println(s1.id+" "+s1.name);

System.out.println(s2.id+" "+s2.name);

}

}

## Output:



## Program 4: function overriding

class Vehicle

{

void run()

{

System.out.println("vehicle is running");

}

}

class Bike extends Vehicle

{

void run()

{

System.out.println("vehicle is running safety");

}

public static void main(String args[])

{

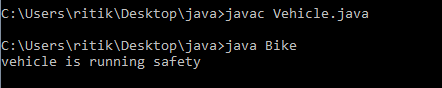
Bike obj = new Bike();

obj.run();

}

}

### Output:



## Program 5: function overloading

class Testoverload

### {

public static void main(String args[])

{

System.out.println("main with syring");

}

public static void main(String args)

{

System.out.println("hello");

}

public static void main()

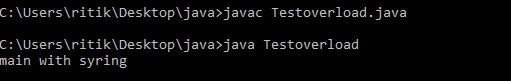
{

System.out.println("hello students");

}

}

### Output:



## Program 6: function overloading

Class Adder

{

static int add(int a,int b)

{

return a+b;

}

static int add(int a,int b,int c)

{

return a+b+c;

}

}

class Testoverloading

{

public static void main(String args[])

{

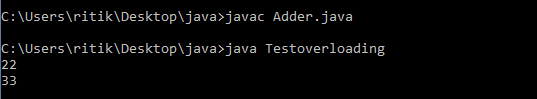
System.out.println(Adder.add(11,11));

System.out.println(Adder.add(11,11,11));

}

}

### Output:



## Program 7: copy constructor

class Students

{

int id;

String name;

Students( int i,String n)

{

id= i;

name =n;

}

Students(Students s)

{

id=s.id;

name=s.name;

}

void display()

{

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

}

public static void main(String args[])

{

Students s1 = new Students(1,"aman");

Students s2 = new Students(s1);

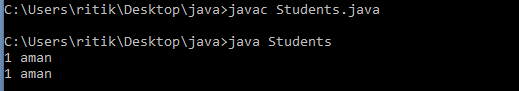
s1.display();

s2.display();

}

}

## Output:



## Program 8: copy constructor

class Std

{

int id;

String name;

Std(int i,String n)

{

id=i;

name=n;

}

Std()

{

}

void display()

{

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

}

public static void main(String args[])

{

Std s1=new Std(101,"aakash");

Std s2=new Std();

s2.id=s1.id;

s2.name=s1.name;

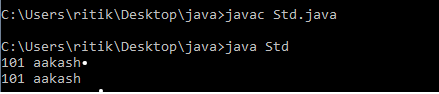
s1.display();

s2.display();

}

}

## Output:



# Program 9 : Abstract class

abstract class Bike

{

Bike()

{

System.out.println("constructor bike created ");

}

abstract void run();

void Changegear()

{

System.out.println("geer changed");

}

}

class Honda extends Bike

{

void run()

{

System.out.println("running safety ");

}

}

class TestAbst {

public static void main (String args[])

{

Bike obj = new Honda ();

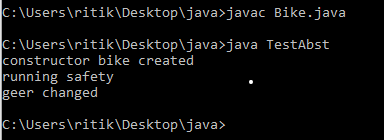
obj.run();

obj.Changegear();

}

}

## Output:



# PROGRAM 10 : SUPER KEYWORD

class Animal

{

Animal()

{

System.out.println("constructor created");

}

String color = "white";

void Eat()

{

System.out.println("eating");

}};

class D extends Animal

{

D()

{

super();

System.out.println(" D constructor ");

}

String color = "black";

void printcolor()

{

System.out.println(color);

System.out.println(super.color);

}

void Eat()

{

System.out.println("eating & eating ");

}

void walk()

{

super.Eat();

}

}

class Test

{

public static void main(String args[])

{

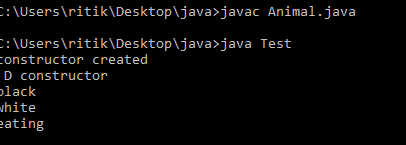
D d = new D();

d.printcolor();

d.walk();

}}

## Output:



# Program 11 : Interface

interface printable

{

void print();

}

interface Showable extends printable

{

void show();

}

class TestInterface implements Showable

{

public void print()

{

System.out.println ("hello");

}

public void show()

{

System.out.println("welcome");

}

public static void main (String args[])

{

TestInterface obj = new TestInterface();

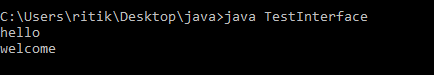
obj.print();

obj.show();

}

}

## Output:



# Program 12: interface

interface printable

{

void print();

}

class A6 implements printable

{

public void print()

{

System.out.println("hello");

}

public static void main ( String args[])

{

A6 obj = new A6();

obj.print();

}

}

## Output:

