1)Java code with class acad and method main to find the sum of two hard-coded integers

Code :

class acad

{

public static void main(String args[])

{

int a=50,b=50;

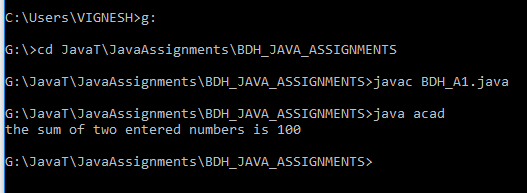
int c=a+b;

System.out.println("the sum of two entered numbers is "+c);

}

}

Output:



2) Above code but input being provided by user :

Code :

import java.util.Scanner;

class acad

{

public static void main(String args[])

{

int a,b,c;

Scanner input=new Scanner(System.in);

System.out.println("Enter two number that you want added one after another:\n");

a=input.nextInt();

b=input.nextInt();

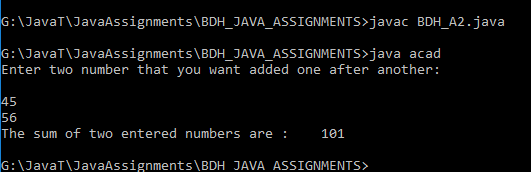
c=a+b;

System.out.println("The sum of two entered numbers are :\t"+c);

}

}

Output :



3) Program with method sum() that accepts 2 parameters from user and print the sum of those 2

Code :

import java.util.Scanner;

import java.io.\*;

class add

{

private int result;

void sum(int x,int y)

{

result=x+y;

}

public static void main(String args[])

{

int a,b;

Scanner s1=new Scanner(System.in);

System.out.println("First number is:\n");

a=s1.nextInt();

System.out.println("Second number is:\n");

b=s1.nextInt();

add sa=new add();

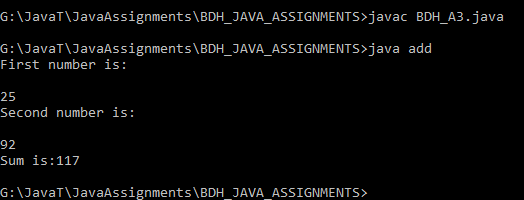
sa.sum(a,b);

System.out.println("Sum is:"+sa.result);

}

}

Output :



4 ) Program that accepts two numbers from stdin and displays odd and even numbers in between them

Code :

import java.util.Scanner;

class Problem1 {

public static void oddEvenList(int no1,int no2){

int min;

int max;

int temp;

if(no1>no2){

min=no2;

max=no1;

}

else if(no1<no2){

min=no1;

max=no2;

}

else{

min=no1;

max=no1;

}

System.out.println("Even Numbers :");

temp=min;

while(temp<=max){

if(temp%2==0){

System.out.print(temp+" ");

temp=temp+2;

}

else{

temp++;

}

}

System.out.println("\nOdd Numbers :");

temp=min;

while(temp<=max){

if(temp%2!=0){

System.out.print(temp+" ");

temp=temp+2;

}

else{

temp++;

}

}

}

public static void main(String args[]){

Scanner scanner=new Scanner(System.in);

int no1;

int no2;

System.out.println("Enter the two numbers:");

no1=scanner.nextInt();

no2=scanner.nextInt();

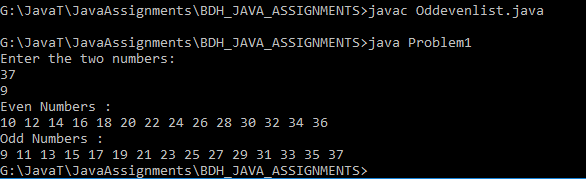
scanner.close();

oddEvenList(no1,no2);

}

}

Output :



5) Program to find the first 10 multiples of number entered:

Code :

import java.util.\*;

class A

{

public static void main(String args[])

{

Scanner input=new Scanner(System.in);

int a,i;

System.out.println("Enter the number for which you want to find the first 10 multiples");

a=input.nextInt();

for(i=0;i<=10;i++)

{

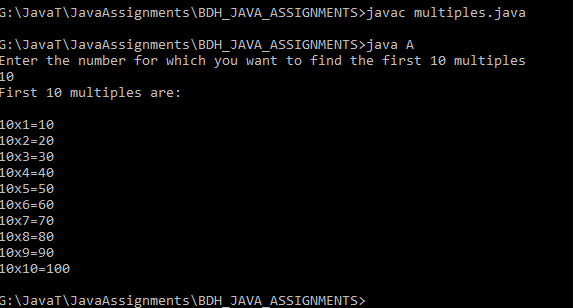
System.out.println(+a+"x"+i+"="+a\*i);

}

}

}

Output :



6) Concept of overloading using method sum()

Code :

class Adder

{

static int sum(int a, int b)

{ return a+b;}

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

{ return a+b+c;}

static double sum(int a,int b,double c)

{ return a+b+c; }

}

class MethodOverloading

{

public static void main(String args[])

{

System.out.println("First method of type int with 2 int arguments is being called:");

System.out.println(Adder.sum(50,50));

System.out.println("Second method of type int with 3 int arguments is being called:");

System.out.println(Adder.sum(50,50,50));

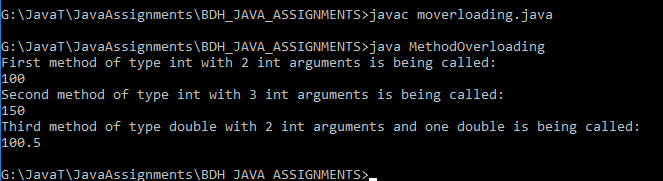
System.out.println("Third method of type double with 2 int arguments and one double is being called:");

System.out.println(Adder.sum(50,50,0.5));

}

}

Output :



7) Can we overload a method with same return type?

Yes , method overloading can be done with same return type. In this case, either the data type of parameters or number of parameters or both should be different. The above code is an example where method overloading has been demonstrated with same return type[method 1 and method 2]

However, what is not possible is to invoke method overloading just my changing the return type alone as the compiler won’t understand which method to call due to ambiguity.

8) Program to sort elements of array in descending order

Code :

import java.util.\*;

class ArraySample

{

public static void main(String args[])

{

int a;

Scanner sc= new Scanner(System.in);

System.out.println("Enter the number of elements you want in the array:");

a=sc.nextInt();

Integer b[]= new Integer[a];

System.out.println("Enter the elements of the array:");

for(int i=0;i<a;i++)

{

b[i]=sc.nextInt();

}

Arrays.sort(b,Collections.reverseOrder());

System.out.println("Elements sorted in descending order:");

for(int j=0;j<b.length;j++)

{

System.out.println(b[j]);

}

}

}

Output :

