1) Write a java code with the class named ‘acad’ and a method ‘main’. Hard Code the program

with two integers and print the sum of those two.

**public** **class** Acad

{

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

{

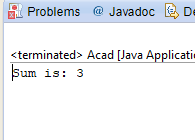
**int** a=1;

**int** b=2;

System.*out*.println("Sum is: "+ (a+b));

}

}



2) Rewrite the above code, where, inputs are provided by the user at runtime and the output is

printed.

**import** java.util.Scanner;

**public** **class** Acad

{

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

{

**int** a;

**int** b;

System.*out*.println("enter values of a and b");

Scanner sc=**new** Scanner(System.*in*);

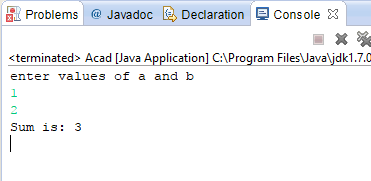
a=sc.nextInt();

b=sc.nextInt();

System.*out*.println("Sum is: "+ (a+b));

}

}



3) Write a program with method name sum() that accepts two parameters from user and print

the sum of two numbers. Output format should be as:

First number is:

Second number is:

Sum is:

**import** java.util.Scanner;

**public** **class** Acad

{

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

{

Acad acad= **new** Acad();

**int** a;

**int** b;

System.*out*.println("enter values of a and b");

Scanner sc=**new** Scanner(System.*in*);

a=sc.nextInt();

b=sc.nextInt();

System.*out*.println("First number is: "+a);

System.*out*.println("Second number is: "+b);

System.*out*.println("Sum is: "+ acad.sub(a,b));

}

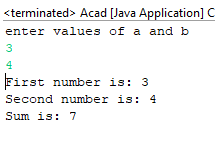
**protected** **int** sub(**int** a, **int** b)

{

**return** a+b;

}

}



4) Write a program to accepts two numbers from stdin and find all the odd as well as even

numbers present in between them.

**import** java.util.Scanner;

**public** **class** Acad

{

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

{

Acad acad= **new** Acad();

**int** a;

**int** b;

System.*out*.println("enter values of a and b");

Scanner sc=**new** Scanner(System.*in*);

a=sc.nextInt();

b=sc.nextInt();

acad.printOddEven(a,b);

}

**protected** **void** printOddEven(**int** a, **int** b)

{

a++;

**while**(a!=b)

{

**if**(a%2==0)

{

System.*out*.println("Even: "+a);

}

**else**

{

System.*out*.println("odd: "+a);

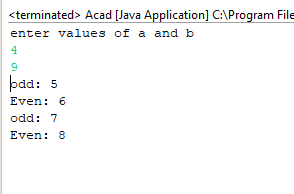
}

a++;

}

}

}



5) Joe is scared to go to school. When her dad asked the reason, joe said she is unable to

complete the task given by her teacher. The task was to find the “first 10 multiples” of the

number entered from stdin . Eg:

Input: 3

O/p:

3 x 1 = 3

3 x 2 = 6

.........

....

3 x 10 = 30

Help Joe in completing the task!

**import** java.util.Scanner;

**public** **class** Acad

{

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

{

System.*out*.println("Enter number");

Scanner sc=**new** Scanner(System.*in*);

**int** a=sc.nextInt();

*printTable*(a);

}

**private** **static** **void** printTable(**int** a) {

**for**(**int** i=1; i<=10;i++)

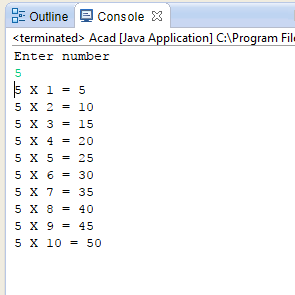
{

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

}

}

}



6) Write a program consisting method sum() and demonstrate the concept of method

overloading using this method.

**public** **class** Acad

{

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

{

System.*out*.println("adding two number "+*sum*(5,10));

System.*out*.println("adding three numbers "+*sum*(5,10,2));

}

**private** **static** **int** sum(**int** i, **int** j, **int** k) {

// **TODO** Auto-generated method stub

**return** i+j+k;

}

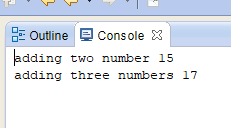
**private** **static** **int** sum(**int** i, **int** j) {

// **TODO** Auto-generated method stub

**return** i+j;

}

}



7) Can you overload a method with same return type.? Explain your answer with proper logic.

In a class, there can be several methods sharing the same name but differ in

1. Parameter types
2. Number of parameters
3. Order of the parameters declared in the method

By depending on the parameters provided for the method, in the run time, compiler determines which version of the method to execute.

An overloaded method may or may not have different return types. But return type alone is not sufficient for the compiler to determine which method is to be executed at run time.

8) Write a program in java using Arrays, that sorts the element in descending order.

**public** **class** Acad

{

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

{

Acad acad= **new** Acad();

**int** a[]={7,4,12,10,11,20};

acad.sortDescending(a);

}

**private** **void** sortDescending(**int**[] a) {

**for**(**int** i=0; i<a.length-1; i++)

{

**for**(**int** j=0;j<a.length-i-1;j++)

{

**if**(a[j]<a[j+1])

{

**int** c=a[j];

a[j]=a[j+1];

a[j+1]=c;

}

}

}

**for** (**int** i : a) {

System.*out*.print(i+" ");

}

}

}

