Session 4

Assignment 1

|  |  |
| --- | --- |
| **Prepared For:** | AcadGild |
|  |  |
| **Document Approval:** | **AcadGild** |
|  |  |
|  |  |
|  |  |
|  |  |
| **Project Title:** | Session 4 – Assignment 1 |
|  |  |
| **Prepared By:** | Duncan Burgess |
|  |  |
|  | dburgess@duncb.com |
|  |  |
| **Primary Engineer:** | Duncan Burgess |
|  |  |
| **Document Reference:** | **Session 4 – Assignment 1** |
|  |  |
| **Start Date:** | 05/09/2017 |
|  |  |
|  |  |

# 

# Contents

[Contents 2](#_Toc492590917)

[Change History 3](#_Toc492590918)

[1. Problem Statement 4](#_Toc492590919)

[2. Solutions 5](#_Toc492590920)

# Change History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Revision** | **Date** | **Authored By** | **Authorised By** | **Sections Affected** | **Reason for Change** |
| Rev 01 | 07/09/2017 | Duncan Burgess |  | All | Initial release. |
| Rev 02 | 11/09/2107 |  |  | Task | Completed assignments |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Problem Statement

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.

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

printed.

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:

4) Write a program to accepts two numbers from stdin and find all the odd as well as even numbers present in between them.

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!

6) Write a program consisting method sum() and demonstrate the concept of method overloading using this method.

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

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

# Solutions

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 arng[]){

int num1 = 12;

int num2 = 25;

int sum = num1 + num2;

System.out.println ("The first Number is " + num1 + " The Second Number is " + num2);

System.out.println ("The Sum is " + (sum));

}

}

**Result**

The first Number is 12 The Second Number is 25

The Sum is 37

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 acad2 {

public static void main (String[] args){

Scanner n1 = new Scanner (System.in);

System.out.println ("Enter a Number1");

int num1 = n1.nextInt();

System.out.println ("Enter a Number2");

int num2 = n1.nextInt();

System.out.println ("The first Number is " + num1 + " The Second Number is " + num2);

System.out.println ("The Sum is " + (num1 + num2));

}

}

**Result**

Enter a Number1

78

Enter a Number2

45

The first Number is 78 The Second Number is 45

The Sum is 123

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 AditionSum {

//Class group{

public static void main(String arng[]){

Scanner data = new Scanner(System.in);

System.out.println("Enter 1st number");

int num1 = data.nextInt();

System.out.println("Enter 2nd number");

int num2 = data.nextInt();

sum(num1,num2);

}

// sum method

public static String sum(int numx, int numy){

int sum=numx+numy;

System.out.println("The First Number is " + numx);

System.out.println("The Second number is " + numy);

System.out.println("The Sum is "+sum);

Return null;

}

}

**Result**

Enter 1st number

47

Enter 2nd number

12

The First Number is 47

The Second number is 12

The Sum is 59

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 OddEven {

public static void main (String[] args){

Scanner input = new Scanner (System.in);

System.out.println("Please enter your 2 numbers");

int n1 = input.nextInt();

int n2 = input.nextInt();

String even = "";

String odd = "";

for(int i = n1; i < n2+1; i++){

if (i % 2 == 0)

even += i + " ";

}

for(int j =n1; j < n2+1; j++){

if (j % 2 != 0)

odd += j + " ";

}

//}

System.out.println("Even Numbers between " + n1 + " and " + n2 + " is " + even);

System.out.println("Odd Nubmers between " + n1 + " and " + n2 + " is " + odd);

}

}

**Result**

Please enter your 2 numbers

7

21

Even Numbers between 7 and 21 is 8 10 12 14 16 18 20

Odd Nubmers between 7 and 21 is 7 9 11 13 15 17 19 21

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 MultiplyByTen {

public static void main (String[] args){

Scanner input = new Scanner (System.in);

System.out.println("Please enter to be multiplied ");

int usernum = input.nextInt();

System.out.println ("Your Number was " + usernum);

for (int i = 1; i < 11; i = i+ 1){

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

}

}

}

**Result**

Please enter to be multiplied

5

Your Number was 5

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

6) Write a program consisting method sum() and demonstrate the concept of method overloading using this method.

public class Overload {

public static void main (String[] args){

int x = multiply (2,6);

int y = multiply (2,6);

System.out.println(x + " and " + y );

}

public static int multiply(int n1, int n2){

return n1 \* n2;

}

public static int multiply( int n1, int n2){

return n1 \* n2;

}

}

Results in error same return tpe.

public class Overload {

public static void main (String[] args){

int x = multiply (2,6);

double y = multiply (2.5,6);

System.out.println(x + " and " + y );

}

public static int multiply(int n1, int n2){

return n1 \* n2;

}

public static double multiply(double n1, double n2){

return n1 \* n2;

}

}

**Result**

12 and 15.0

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

Method signature only includes the name of the method + input parameter(s).

So the return type of the method is not included in the method signature.

On the other hand overloading means:

Having two or more methods in a class with the exact same name but different parameter types or different number of parameters or both.

Compiler does not indicate the return types of your methods, so cannot understand the difference between two show methods with two int parameters. Same mistake happens for your two show methods without parameters.

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

**import** java.util.Arrays;

**public** **class** Sorting\_Num {

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

//Array of Integers

**int** numbers[] = {20,17,29,35,7,74,-2};

System.***out***.println("Before Sorting");

**for** (**int** n1 : numbers){

System.***out***.print(" " + n1);

}

System.***out***.println();

// sorting array

Arrays.*sort*(numbers);

// Sort Ascending

System.***out***.println("After sorting - Increasing");

**for** (**int** n1 : numbers) {

System.***out***.print(" " + n1);

}

//Reverse Sorting - Sort Descending

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

*reverse*(numbers);

System.***out***.println("After Sorting - descending order ");

**for** (**int** n1 : numbers) {

System.***out***.print( " " + n1);

}

//Method to reverse

**public** **static** **void** reverse(**int**[] input) {

**int** last = input.length - 1;

**int** middle = input.length / 2;

**for** (**int** i = 0; i <= middle; i++) {

**int** temp = input[i];

input[i] = input[last - i];

input[last - i] = temp;

}

}

}

**Result**

Result Before Sorting

20 17 29 35 7 74 -2

After sorting - Increasing

-2 7 17 20 29 35 74

After Sorting - descending order

74 35 29 20 17 7 -2

Note in real environment the array would have been copied before modification to retain originality.