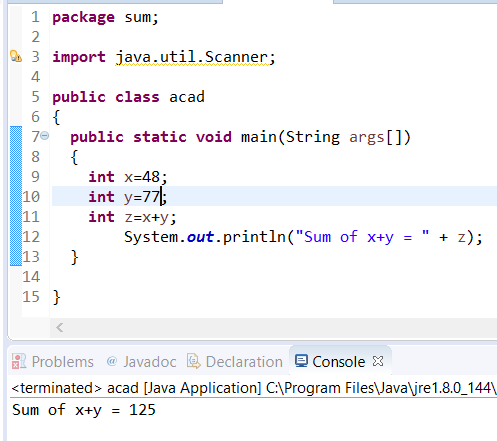
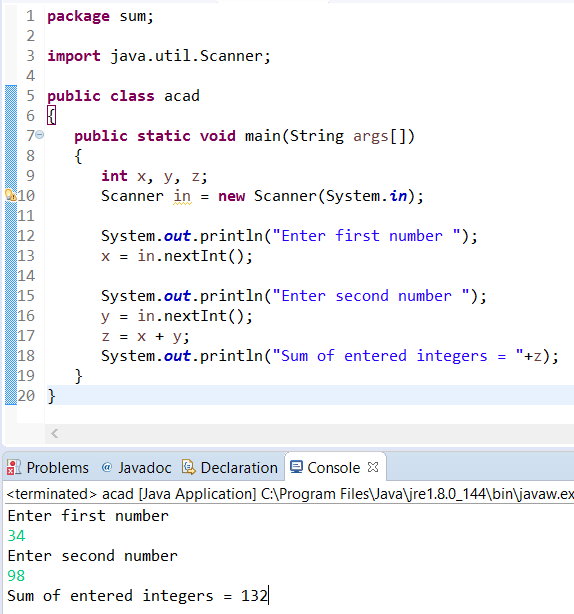
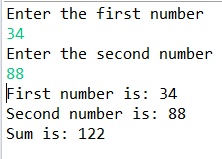
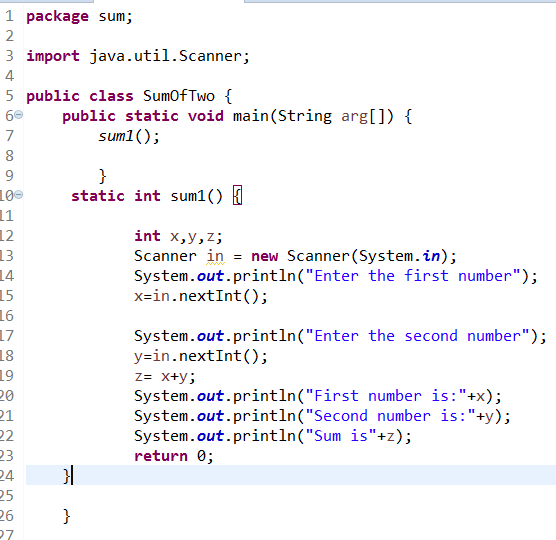
**Assignment 4.1**

1. **Code to get the sum of 2 numbers:**

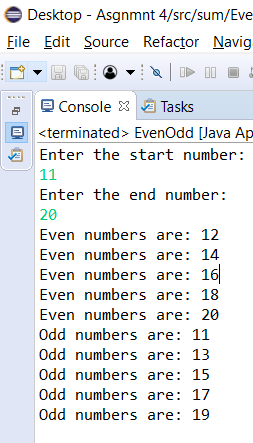


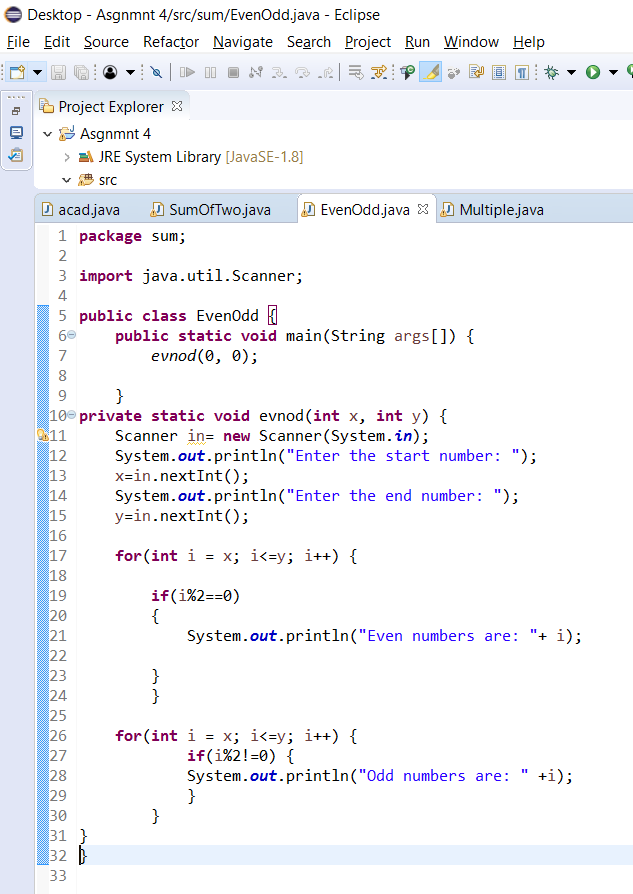
1. **Sum of 2 numbers by asking the input from the user:**
2. **Sum of 2 numbers with the method sum()**



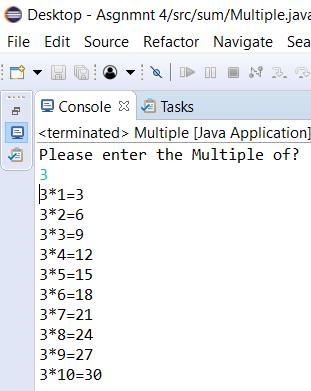


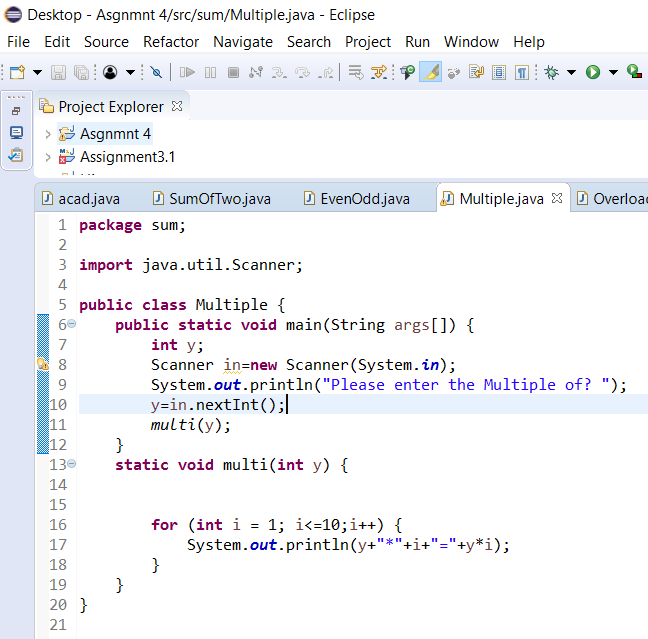
1. **Print Odd and Even numbers:**





1. **Helping Joe with the homework**





1. **Program for overloading:**

**package** sum;

**import** java.util.Scanner;

**public** **class** OverloadingMethodOfSum {

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

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

**int** x=0,y=0,z=0;

**boolean** i;

i=**true**;

**while**(i){

**int** ch;

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

System.***out***.println("Enter the Number to add");

x=in.nextInt();

System.***out***.println("Enter the Second Number to add");

y=in.nextInt();

System.***out***.println("Do you Want to add another number? if Yes hit 1 otherwise 0 for NO");

ch=in.nextInt();

**if**(ch == 1)

{

System.***out***.println("Enter the Third Number to add");

z=in.nextInt();

*sum*(x,y,z);

}

**else**

{

*sum*(x,y);

}

**break**;

}

}

**static** **void** sum(**int** x, **int** y)

{

**int** add=x+y;

System.***out***.printf("The Sum of x = "+x+" and y = "+y+" is "+add);

}

**static** **void** sum(**int** x, **int** y,**int** z)

{

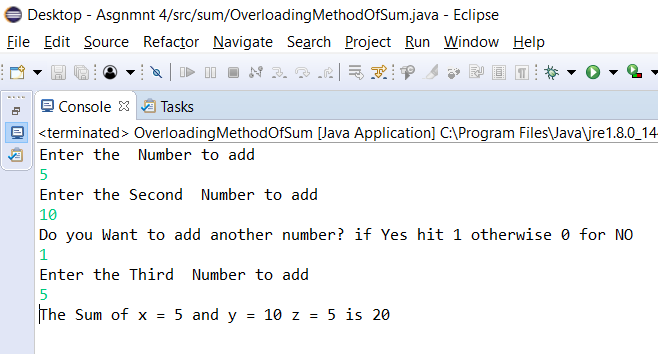
**int** add=x+y+z;

System.***out***.printf("The Sum of x = "+x+" and y = "+y+" z = "+z+" is "+add);

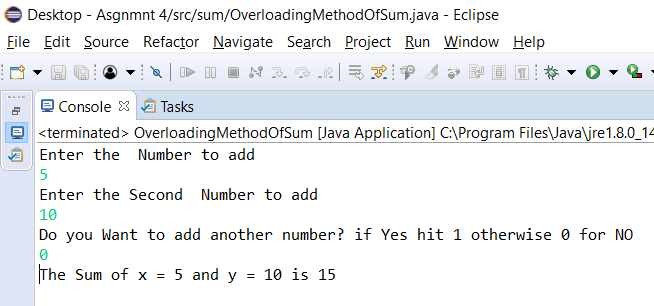
}

}

Output with only 2 numbers:



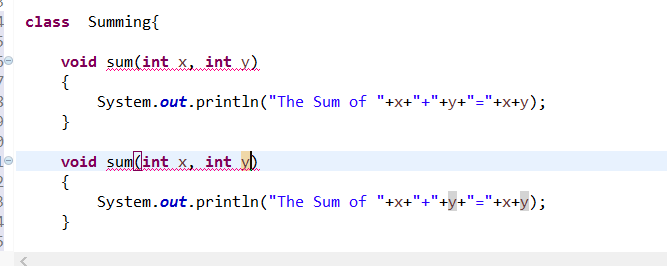
Output with only 3 numbers:

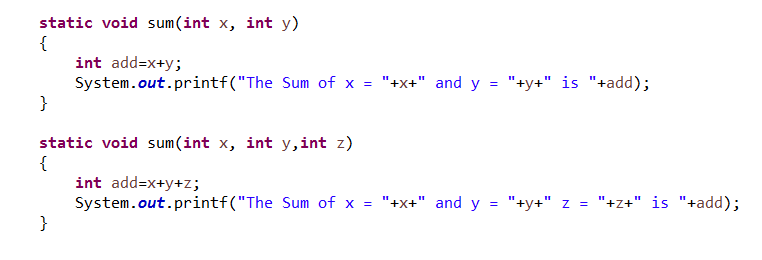


1. **Can you overload a method with same return type?**

Ans: We can overload a method with the same return type **PROVIDED** we have different parameters. As you can see in the following screen shots, we have the same parameters and the same return type hence we get an error (*red line*)

In the second screenshot as you can we, we have same return type, however the parameters are different.





1. **Arrays in descending order:**

