**Adisri Sarode  
  
BASIC DATA STRUCTURE ASSIGNMENT**

[Q1](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2148): Find out if the given number is an Armstrong number or not.

**Description :-**

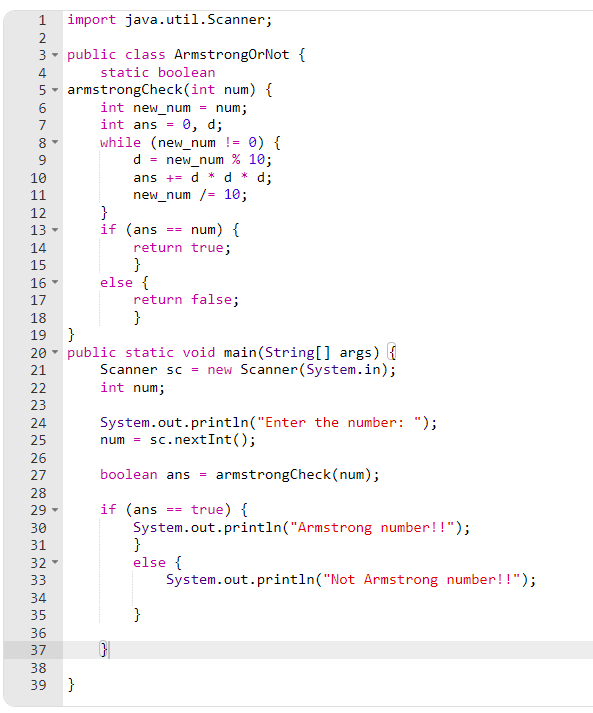
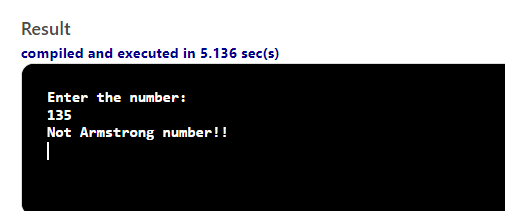
An Armstrong number of three digits is an integer, where the sum of the cubes of its digits is equal to the number itself.

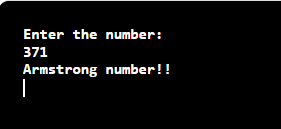
Consider the example: 371=> 3^3 + 7^3 + 1^3 = 371 ( If you add those all numbers, the final digit should be same as given number ).

**Specifications:**

class ArmstrongOrNot {  
    public boolean armstrongCheck(int num) {}

public class Assignment1Q1 {  
    public static void main(String[] args) {}  
}

**  
  
**

****

Q2. Find out all the Armstrong numbers falling in the range of 100-999

Description :-

An Armstrong number of three digits is an integer, where the sum of the cubes of its digits is equal to the number itself.

Consider the example: 371=> 3^3 + 7^3 + 1^3 = 371 ( If you add those all numbers, the final digit should be same as given number ).

Find the Armstrong numbers between 100 to 999.

**Test cases:**

Output : 153 370 371 407

**Specifications:**

class ArmstrongNumBetweenRange{

public int[] armstrongNumbersInRange(int min , int max){}

}

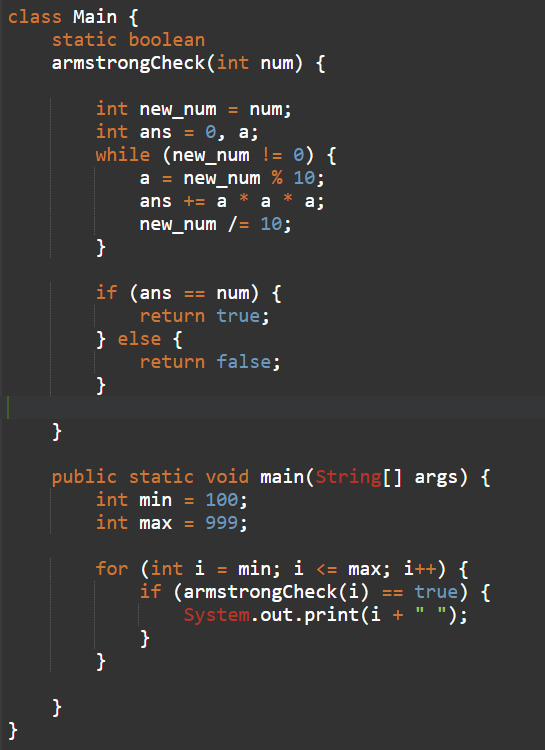
public class Assignment1Q2 {

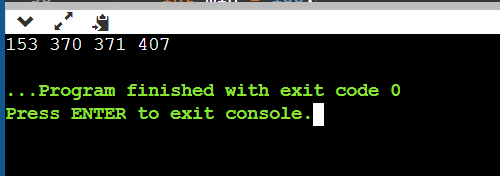
public static void main (String [] args) {

int min = 100;int max = 999;

}

}

**  
  
OUTPUT:**

****

Q3. Find out the simple as well as the compound interest of supplied value

Description:-

Simple Interest:- Generally, simple interest paid or received over a certain period is a fixed percentage of the principal amount that was borrowed or lent

Simple Interest = (P×r×n)/100

where:

P = Principal amount

r = Annual interest rate

n = Term of loan, in years

Compound Interest:-Compound interest accrues and is added to the accumulated interest of previous periods; it includes interest on interest, in other words.

Compound Interest = P(1+r)^t-P

Where:

P=Principal amount

r=Annual interest rate

t=Number of years interest is applied

**Specifications:**

class SiCi {

public double simpleInterest(double principalAmount,int time,double interestRate){}

public double compoundInterest(double principalAmount,int time,double interestRate){}

}

public class Assignment1Q3 {

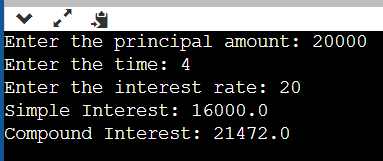
public static void main (String args[]) {}

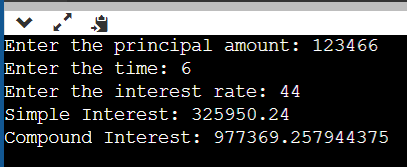
}

****

**OUTPUT:**

**Test case 1**

****

**Test Case 2  
**

[**Q4**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2262)**. Supply marks of three subject and declare the result, result declaration is based on below conditions:**

**Condition 1: -All subjects marks is greater than 60 is Passed**

**Condition 2: -Any two subjects marks are greater than 60 is Promoted**

**Condition 3: -Any one subject mark is greater than 60 or all subjects’ marks less than 60 is failed.**

**Description:-**

Specify the marks of 3 subjects and the results will be declared based on the conditions above and for reference go through the test cases for better understanding.

**Test cases:-**

**TestCase1:-**

                            Input:-     10          10          10

                            Output:-  failed

**TestCase2:-**

                            Input:-      70          10          10

                            Output:-   failed

**TestCase3:-**

                            Input:-      10          20          40

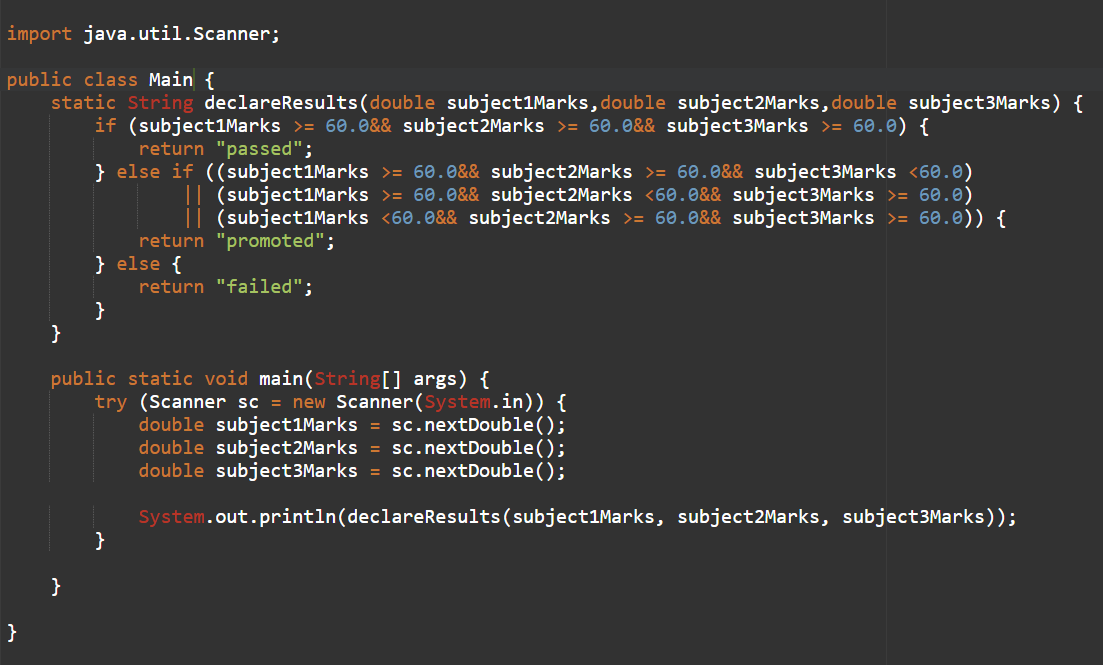
                            Output:-   passed

**TestCase4:-**

                             Input:-      10          30          40

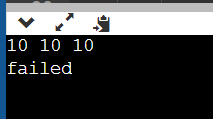
                            Output:-    Passed

                                            promoted

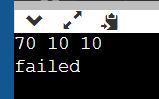
****

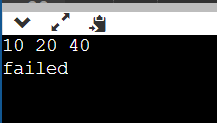
**OUTPUT:**

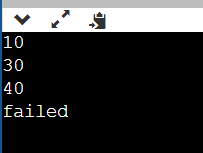
**Test Case 1**

****

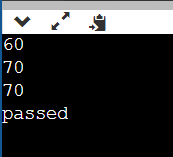
**Test Case 2**

****

**Test case 3  
**

**Test Case 4  
**

**Test Case 5**

****

[**Q5**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2153)**. Calculate the income tax on the basis of following table.**

**Note:-Assume slab is consider for Male, Female as well as Senior citizen**

**Slab                                   Income Range                             Tax payable in Percentage**

**Slab A                               0-1,80,000                                                                  Nil**

**Slab B                               1,81,001-3,00,000                                                    10%**

**Slab C                               3,00,001-5,00,000                                                    20%**

**Slab D                               5,00,001-10,00,000                                                 30%**

**Accept CTC from user and display tax amount**

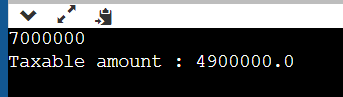
**Description:-**

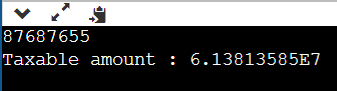
Given 4 different types of slabs along with the percentage of tax payable in association with income ranges which are applicalble to Male,Female as well as Senior citizen.You need to specify the CTC to display the taxable amount using the above slab rates.

****

**Output:**

**Test Case 1:**

****

**Test Case 2: **

[**Q6**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2170)**. Consider a CUI based application, where you are asking a user to enter his Login name and password, after entering the valid user-id and password it will print the message “Welcome” along with user name. As per the validation is concerned, the program should keep a track of login attempts. After three attempts a message should be flashed saying “Contact Admin” and the program should terminate.**

**Description:-**

You have to create a CUI based application in which you will declare and initialize the userId and password.You have to enter the credentials when you login to the application(when you run the program it should ask the user to enter the credentials),if the entered credentials are validated correctly you should see a welcome message with the userId or else you should have total 3 attempts to enter the correct credentials.If you fail to enter the right credentials in your 3rd attempt you should display a message "Contact Admin".

**Test Cases:-**

Sample Input:- (as per Specifications)

userId = "Ajay",password = "password";

SampleOutput:-

**Example1:-**

Enter userId

ajay

Enter password

password

You have entered wrong credentials , please enter the right credentials.

**Example2:-**

Enter userId

Ajay

Enter password

pass

You have entered wrong credentials ,please enter the right credentials.

Example3:-

Enter userId

Ajay1

Enter password

password

You have entered wrong credentials 3 times

Contact Admin

Example4:-

Enter userId

Ajay

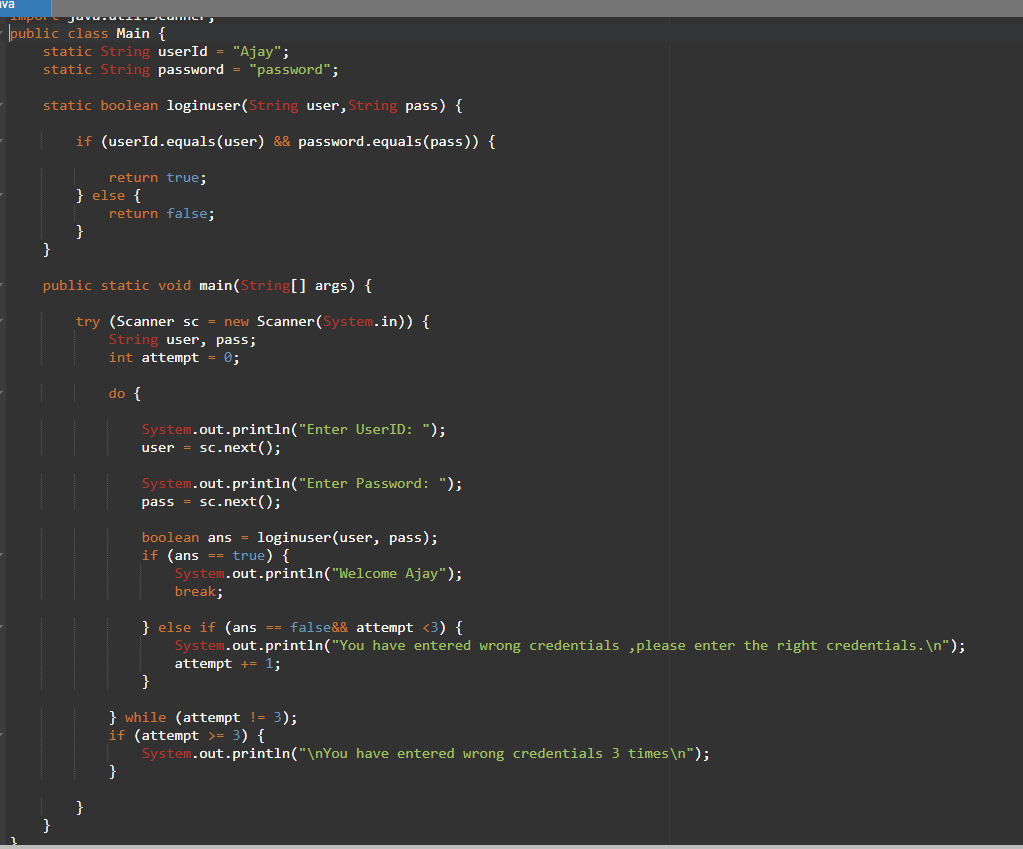
Enter password

password

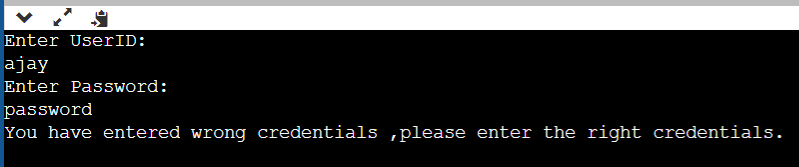
Welcome Ajay

**Specifications:-**

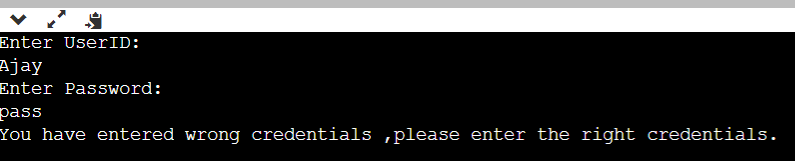
class Login{  
    String userId = "Ajay",password = "password";  
    public String loginUser(String user, String pass) {}

}  
public class Assignment1Q6 {  
    public static void main(String[] args) {}  
}  
  
  
  
  
**OUTPUTS:**

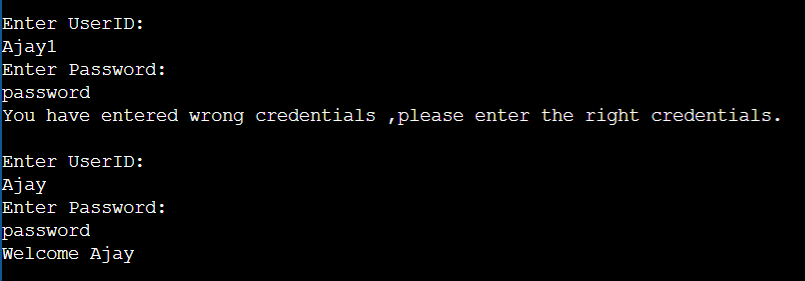
**Test Case 1:**

****

**Test Case 2:**

****

**Test Case 3 & 4:**

****

[**Q7**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2169)**. There is an Array which is of the size 15, which may or may not be sorted. You should write a program to accept a number and search if it in contained in the array**

**Example:**

**5 12 14 6 78 19 1 23 26 35 37 7 52 86 47**

**Value to be search is 19**

**Description:-**

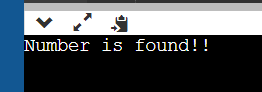
Given an array, the task is to check whether a certain element is present in this given Array or not.

**Specifications-**

class SearchArray{  
    public boolean searchArray(int[] arr,int toCheckValue){}  
}  
public class Assignment1Q7 {  
    public static void main(String[] args) {  
        int arr[] = { 5,12,14,6,78,19,1,23,26,35,37,7,52,86,47};  
        int valueToCheck = 19;  
}

****

**Output:**

****

[**Q8**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2168)**. Using the below table write method apply sorting using Bubble Sort.**

**Example:**

**5 12 14 6 78 19 1 23 26 35 37 7 52 86 47**

**Description:-**

Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of Ο(n2) where n is the number of items.

**Example-**

              Input :          5 12 14 6 78 19 1 23 26 35 37 7 52 86 47

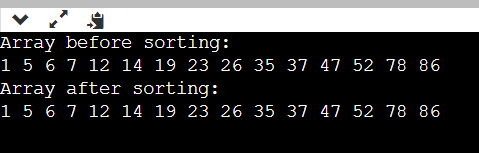
              Output:       1 5 6 7 12 14 19 23 26 35 37 47 52 78 86

**Specifications:**

class BubbleSort{  
  
    public int[] bubbleSort(int arr[]) {}  
}  
public class Assignment1Q8 {  
    public static void main(String args[]) {  
        int arr[] = {5,12,14,6,78,19,1,23,26,35,37,7,52,86,47};  
    }  
}

****

**Output:**

****

[**Q9**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2159)**. Accept the marks of three students for the subject say A, B, C. Find the total scored and the average in all the subjects. Also Find the Total and Average scored by students in each respective Subject.**

**Description:-**

Enter the marks of 3 students for subjects A,B,C. Find the total marks secured by respective student in all the subjects and also find the total and average scored by students subject wise.

Example:-

              Sample Input:-

                             marks of Student 1 in subjects A,B,C

                                           10 20 30

                             marks of Student 2 in subjects A,B,C

                                           10 20 30

                             marks of Student 3 in subjects A,B,C

                                           10 20 30

              Sample Output:-

                             180                                    //Total marks of all the students in all subjects

                             60.0                                   //Average marks of all the students in all subjects

                             30                                       // Total marks scored by students in subject A

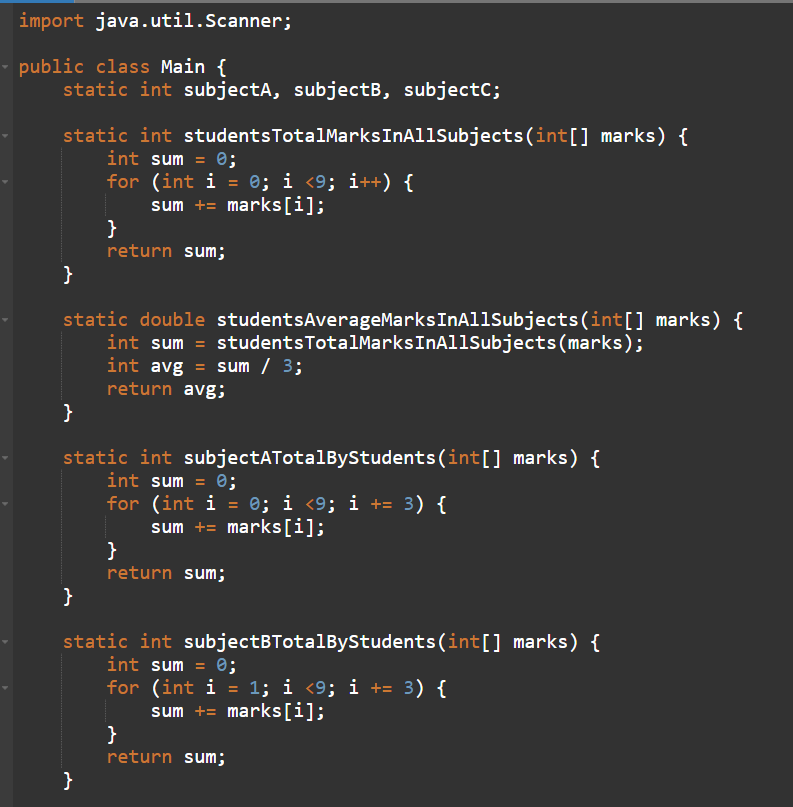
                             10.0                                   // Average marks scored by students in subject A

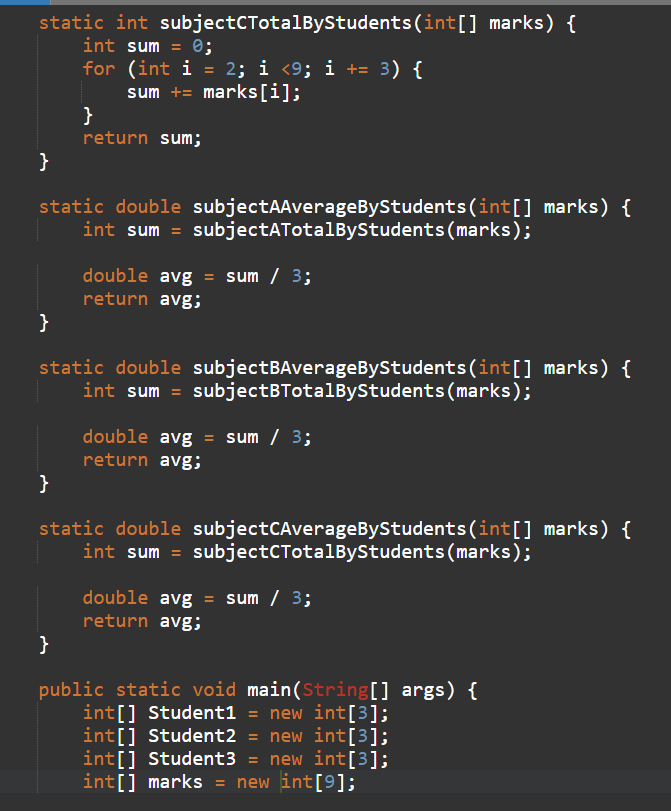
                             60                                       // Total marks scored by students in subject B

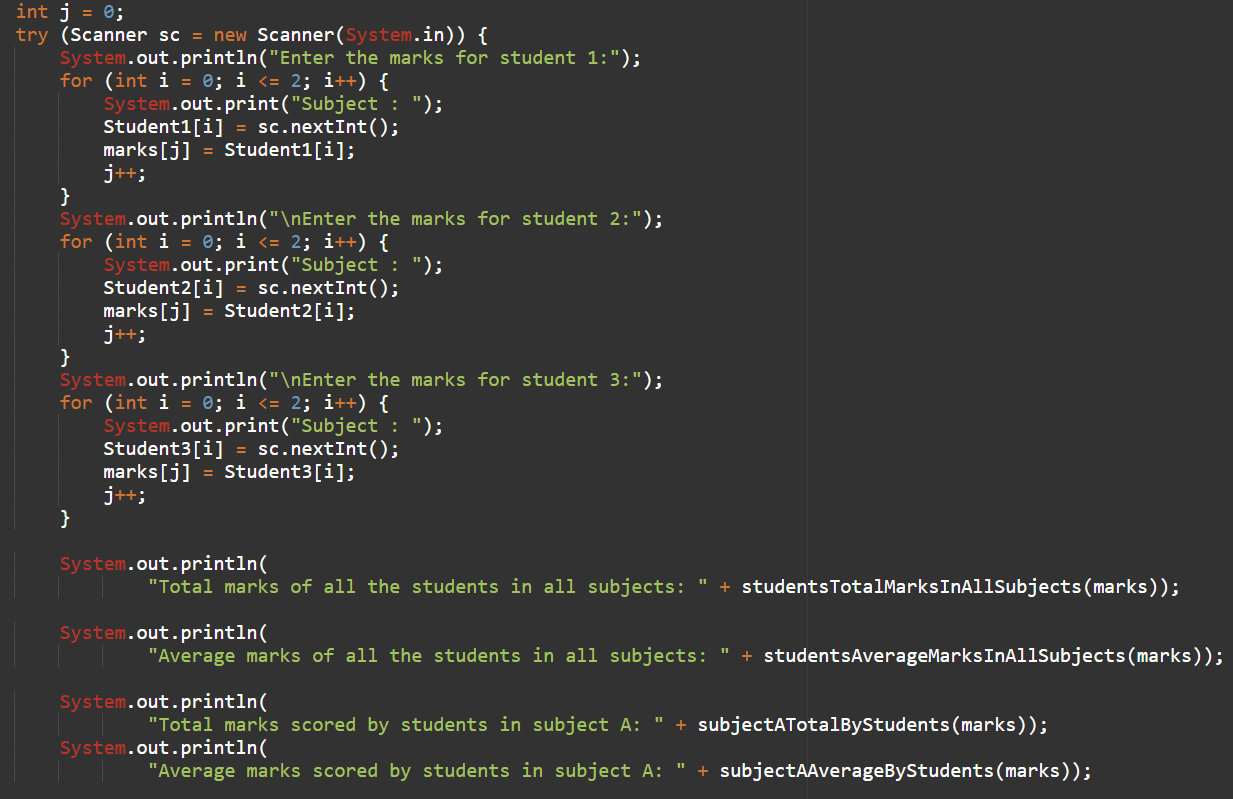
                             20.0                                   // Average marks scored by students in subject B

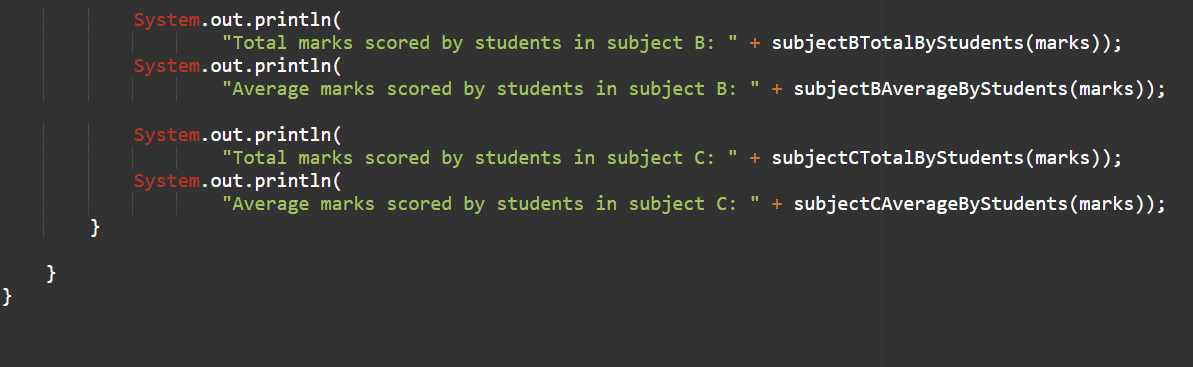
                             90                                       // Total marks scored by students in subject C

                             30.0                                   // Average marks scored by students in subject C

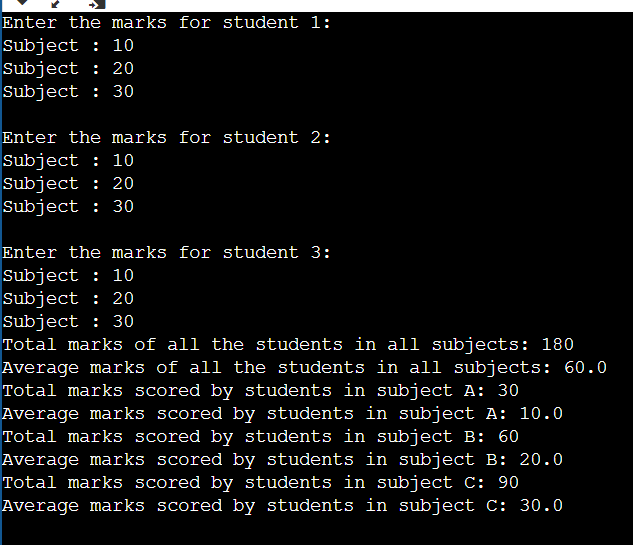
****

****

****

****

**Output:**

****