**Basic Data Structures in Java.:**

Q1: 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) {}

}

**Code:**

**import** java.util.\*;

**public** **class** ArmstrongOrNot {

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

**int** num,a,sum=0,orig;

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

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

num=sobj.nextInt();

orig=num;

**while**(num>0)

{

a=num%10;

sum=sum+(a\*a\*a);

num=num/10;

}

**if**(orig==sum)

{

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

}

**else**

{

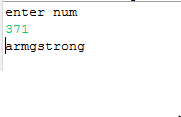
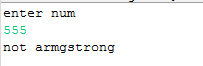
System.***out***.println("not armgstrong");

}

}

}

**Output:**



**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 {

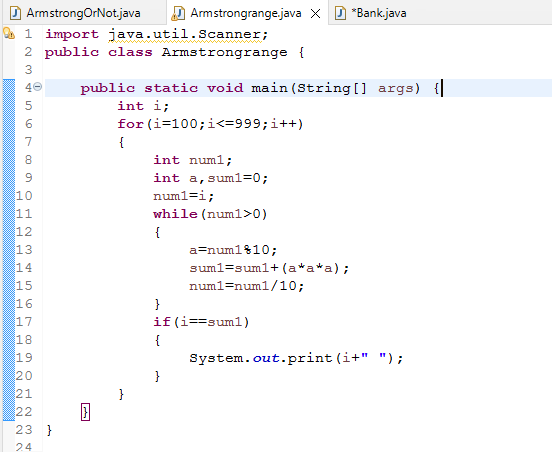
public static void main (String [] args) {

int min = 100;int max = 999;

}

}

**Code:**



**Output:**



[**Q3**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2261)**. 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

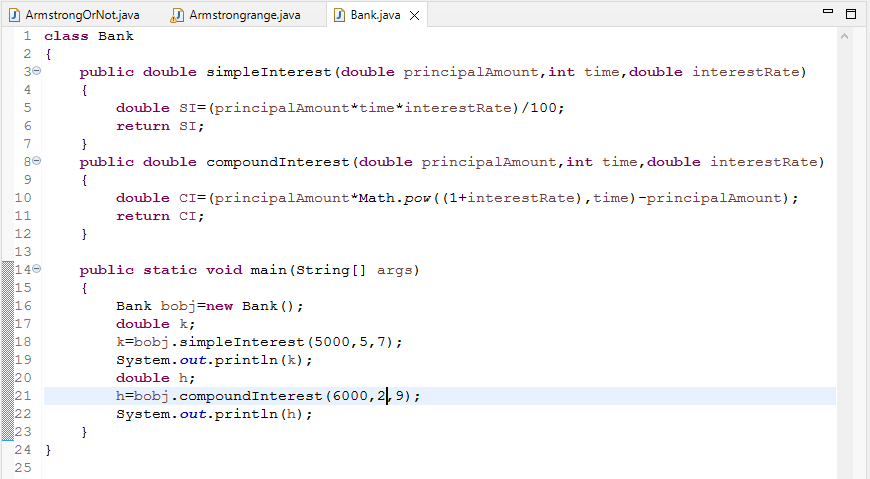
              t=Number of years interest is applied

##### Specifications:

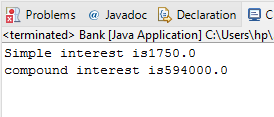
class SiCi {

    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[]) {  
}

**Input**



**output**



[**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

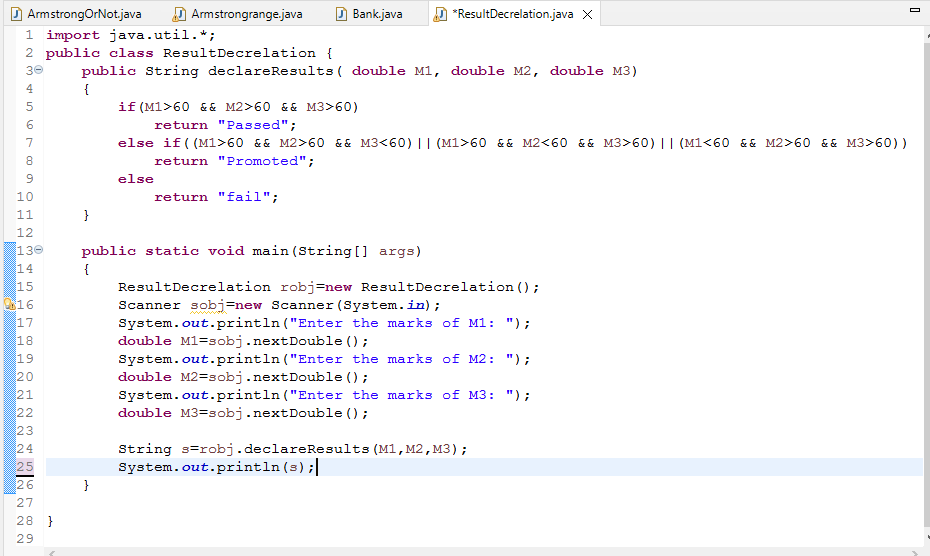
                            Output:-    Passed

                                            promoted

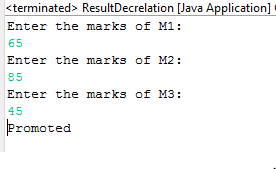
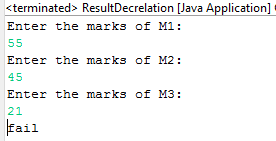
**Specifications:**

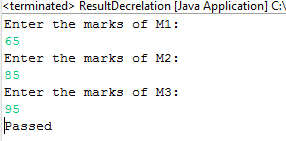
class ResultDeclaration{  
    public String declareResults( double subject1Marks, double subject2Marks, double subject3Marks) {}  
}  
public class Assignment1Q4 {  
    public static void main(String[] args) {}  
}

**Input**



**output**



[**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.

**Specifications:**

class TaxAmount{  
    public double calculateTaxAmount(int ctc){}  
}  
public class Assignment1Q5 {  
    public static void main(String args[]) {}  
}

**input**