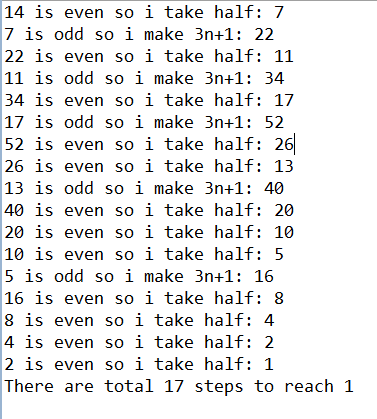
**Assignment 1: Conditional and Loop Statements**

**Problem description:** Write a method which accepts a number as a parameter and then displays the Hailstone sequence for that number. The problem can be expressed as follows:

* Pick some positive integer and call it n.
* If n is even, divide it by two.
* If n is odd, multiply it by three and add one.
* Continue this process until n is equal to one.c

Your program should be able to produce a sample run that looks like this:



**Input:**

Input is a positive number

**Output:**

Output is a series of steps showing how it reached the number and then should return total count of steps

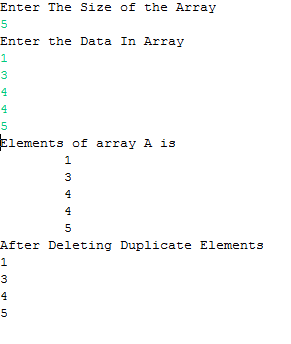
**Assignment 2: 1D Array**

**Problem description:** Delete Duplicate Elements from an Array.

**Note:-**

* **Declare 2 arrays.**
* **Read size of the arrays from the key board.**
* **Enter Data into the First array from the keyboard.**
* **Print the Entered array values in the console.**
* **Write logic to remove duplicate elements from first array and keep it in second array.**
* **Print the array values in console.**

**Sample output:-**



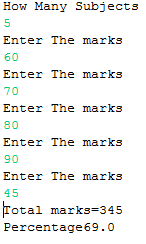
**Assignment 3: 1D Array**

**Problem description:** Write a program which accepts the marks of a student into 1D array from the key board and finds total marks in percentage.

**Note:-**

* **Take the number of subjects from the key board.(Use Scanner for accepting data)**
* **Declare an array of size equal to the number of subjects entered.**
* **Store the marks into an array.**
* **Display the total marks in the console.**
* **Find the percentage of the mark and display in the console.**

**Sample output:-**



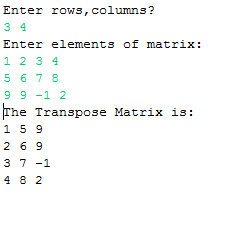
**Assignment 4: 2D Array Implementation**

**Problem description:** Write a program which accepts elements of a matrix and display the transpose.

**Note:-**

* **Accept the rows and columns of the matrix from the user.(Use Scanner for accepting data)**
* **Declare a 2D array int arr[][]=new int[r][c] where r = row and c=column entered by the user .**
* **Accept the matrix from the key board.**
* **Write the logic to find transpose of a matrix and display the result.**

Sample output:-



**Assignment 5: Array**

**Problem description:** Write an application to store ‘N’ numbers of type integers and sort it using below explained logic. Your program should display the sorted numbers in a formatted way

**Example:** Let us take the array of numbers "5 1 4 2 8", and sort the array from lowest number to greatest number using bubble sort. In each step, elements written in **bold** are being compared. Three passes will be required.

**First Pass:**  
( **5** **1** 4 2 8 ) \to ( **1** **5** 4 2 8 ), Here, algorithm compares the first two elements, and swaps since 5 > 1.  
( 1 **5** **4** 2 8 ) \to ( 1 **4** **5** 2 8 ), Swap since 5 > 4  
( 1 4 **5** **2** 8 ) \to ( 1 4 **2** **5** 8 ), Swap since 5 > 2  
( 1 4 2 **5** **8** ) \to ( 1 4 2 **5** **8** ), Now, since these elements are already in order (8 > 5), algorithm does not swap them.  
**Second Pass:**  
( **1** **4** 2 5 8 ) \to ( **1** **4** 2 5 8 )  
( 1 **4** **2** 5 8 ) \to ( 1 **2** **4** 5 8 ), Swap since 4 > 2  
( 1 2 **4** **5** 8 ) \to ( 1 2 **4** **5** 8 )  
( 1 2 4 **5** **8** ) \to ( 1 2 4 **5** **8** )  
Now, the array is already sorted, but our algorithm does not know if it is completed. The algorithm needs one **whole** pass without **any** swap to know it is sorted.  
**Third Pass:**  
( **1** **2** 4 5 8 ) \to ( **1** **2** 4 5 8 )  
( 1 **2** **4** 5 8 ) \to ( 1 **2** **4** 5 8 )  
( 1 2 **4** **5** 8 ) \to ( 1 2 **4** **5** 8 )  
( 1 2 4 **5** **8** ) \to ( 1 2 4 **5** **8** )

**Assignment 6: String handling**

**Estimated time: 15 Mins**

**Problem description:** In an online application, a university would like to validate the university seat number (USN) entered by its student.

Example: A sample USN looks like: **1DS09CS010**

Following are the constraints to enter registration number:

1. Each USN must be length of 10 characters
2. 1st character must be digit and have value either '1' or '2'
3. 2nd and 3rd characters must be upper case letters
4. 4th and 5th characters must be digits and can have values between 0-9
5. 6th and 7th characters must be upper case letters and can have combination any of following substring: CS, IS,EC and ME
6. 8th,9th and 10th characters must be digits and can have values between 0-9

Create class called TestUSN which has main() method to:

* Initialize USN variable with string value
* Validate USN against above constraints
* Print "Success" or "Failure" message based on result of validation

**Assignment 7: Type casting and variable length of arrays**

**Problem description:** Write a program which performs addition of elements which are stored in two arrays of type double. Arrays lengths may be variable in size. The resultant values must be stored in an integer array. Display the resultant integer array in a formatted way.

**Example:**

Input:

|  |  |  |  |
| --- | --- | --- | --- |
| dInputArray1[] | 10.0 | 20.0 | 30.0 |
| dInputArray2[] | 20.0 | 50.0 | 30.0 | 70.0 | 80.0 |

Output:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| iSumArray[] | 10 | 70 | 60 | 70 | 80 |

**Assignment 8: Recursive method**

**Problem description:** Write an application which implements solution for tower of Hanoi problem for ‘n’ discs using recursive method. Here is the explanation about the problem.

**Tower of Hanoi**

The **Tower of Hanoi** is a mathematical game or puzzle. It consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod, obeying the following rules:

* Only one disk may be moved at a time.
* Each move consists of taking the upper disk from one of the rods and sliding it onto another rod, on top of the other disks that may already be present on that rod.
* No disk may be placed on top of a smaller disk.

This puzzle can be solved either by iterative or by recursive.

Recursive solution:

 It can be solved by breaking the problem down into a collection of smaller problems and further breaking those problems down into even smaller problems until a solution is reached.

**For example:**

* label the pegs A, B, C — these labels may move at different steps
* let *n* be the total number of discs
* number the discs from 1 (smallest, topmost) to *n* (largest, bottommost)

To move *n* discs from peg A to peg C:

1. move *n*−1 discs from A to B. This leaves disc *n* alone on peg A
2. move disc *n* from A to C
3. move *n*−1 discs from B to C so they sit on disc *n*

The above is a recursive algorithm: to carry out steps 1 and 3, apply the same algorithm again for *n*−1

**Assignment 9: Array of objects**

**Problem description:** A coffee shop would like to find out the customer feedback rating about its services. The customer class shown below:



**Example:** Assume that the shop will collect feedback from ‘N’ customers. Following are the sample customer feedback values.

Customer 1: 3 out of 5

Customer 2: 4 out of 5

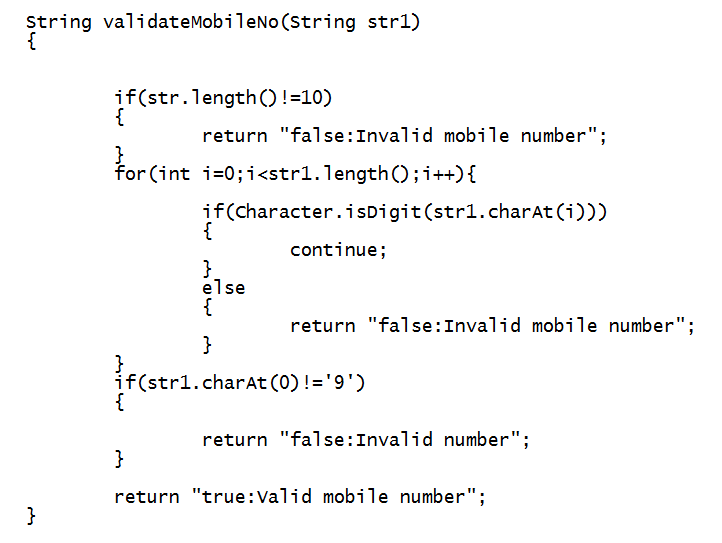
Customer 3: 2.5 out of 5

Write an application which creates array of ‘N’ customer objects to store feedback values of these customers and print the average feedback rating.

**Assignment 10: Unchecked Exceptions**

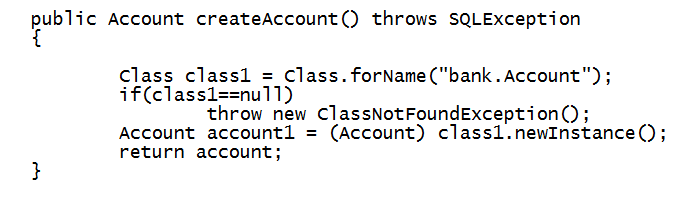
**Problem description:** Below is the third party library function, which helps to process the given string for certain validations.

However this code does not give guarantee of executing always. Why it is so? How can we resolve this issue by using try catch block? Write an application which uses this library function with appropriate try & catch block



**Assignment 11: throw and throws keyword**

**Problem description:** The below code generates an error despite of using ‘throws’ keyword. Observe the code and debug the error. Write a program which invokes the below method.



**Assignment 12: SQL query**

**Problem description:** Given table with the following attributes

|  |  |  |
| --- | --- | --- |
| students | | |
| STUDENT\_NO | SURNAME | FORENAME |
| 20060101 | Dickens | Charles |
| 20060102 | ApGwilym | Dafydd |
| 20060103 | Zola | Emile |
| 20060104 | Mann | Thomas |
| 20060105 | Stevenson | Robert |

|  |  |
| --- | --- |
| modules | |
| MODULE\_CODE | MODULE\_NAME |
| CM0001 | Databases |
| CM0003 | Operating Systems |
| CM0004 | Graphics |
| CM0002 | C Programming |

|  |  |  |
| --- | --- | --- |
| marks | | |
| STUDENT\_NO | MODULE\_CODE | MARK |
| 20060101 | CM0001 | 80 |
| 20060101 | CM0002 | 65 |
| 20060101 | CM0003 | 50 |
| 20060102 | CM0001 | 75 |
| 20060102 | CM0003 | 45 |
| 20060102 | CM0004 | 70 |
| 20060103 | CM0001 | 60 |
| 20060103 | CM0002 | 75 |
| 20060103 | CM0004 | 60 |
| 20060104 | CM0001 | 55 |
| 20060104 | CM0002 | 40 |
| 20060104 | CM0003 | 45 |
| 20060105 | CM0001 | 55 |
| 20060105 | CM0002 | 50 |
| 20060105 | CM0004 | 65 |

Execute following statements in MySQL:

create table students ( student\_no varchar(10), surname varchar(20), forename varchar(20));

create table modules ( module\_code varchar(8), module\_name varchar(20));

create table marks ( student\_no varchar(10), module\_code varchar(8), mark integer);

insert into students values ('20060101','Dickens','Charles');

insert into students values ('20060102','ApGwilym','Dafydd');

insert into students values ('20060103','Zola','Emile');

insert into students values ('20060104','Mann','Thomas');

insert into students values ('20060105','Stevenson','Robert');

insert into modules values ('CM0001', 'Databases');

insert into modules values ('CM0002', 'Programming Languages');

insert into modules values ('CM0003', 'Operating Systems');

insert into modules values ('CM0004', 'Graphics');

insert into marks values ('20060101', 'CM0001', 80);

insert into marks values ('20060101', 'CM0002', 65);

insert into marks values ('20060101', 'CM0003', 50);

insert into marks values ('20060102', 'CM0001', 75);

insert into marks values ('20060102', 'CM0003', 45);

insert into marks values ('20060102', 'CM0004', 70);

insert into marks values ('20060103', 'CM0001', 60);

insert into marks values ('20060103', 'CM0002', 75);

insert into marks values ('20060103', 'CM0004', 60);

insert into marks values ('20060104', 'CM0001', 55);

insert into marks values ('20060104', 'CM0002', 40);

insert into marks values ('20060104', 'CM0003', 45);

insert into marks values ('20060105', 'CM0001', 55);

insert into marks values ('20060105', 'CM0002', 50);

insert into marks values ('20060105', 'CM0004', 65);

commit;

Write SQL statement for following query:

1. Increase the marks by 5 for the module operating system.

2. Update the surname of the students by adding the text “mind” before their surname. Add only the string “mind”, if it has a null value

3. Print student details who have scored more than 60 in C Programming module

4. Select students who have scored more than 80 marks in Database module

**Assignment 13: String Handling**

**Problem description:** Write a method to print the consecutive characters (ignore cases) and the number of times appearing in a String.

public static void printConsecutiveCharacters(String input) {

}

Invoke the above method using main() method.

|  |  |
| --- | --- |
| Sample Input | Sample Output |
| “I saw a CD play-er and a modem in ccd” | CD 2  DE 1 |
| |  |  | | --- | --- | | “Student List do not exist in sys-tem” |  | | ST 4  DE 1  NO 1 |

**Assignment 14: Conditional and Loop statements**

**Problem description:** Triplets are a set of three similar things.

Write a method to print all the triplets <A, B, C> such that A+B = C

public static void printTriplets(int[ ] data) {

}

Invoke the above method using main() method.

|  |  |
| --- | --- |
| Sample Input | Sample Output |
| data ={2,3,4,5,7} | <2,3,5>  <2,5,7>  <3,4,7> |
| data = {1,2,3,4,5,7,9} | <1,2,3>  <1,3,4>  <1,4,5>  <2,3,5>  <2,5,7>  <3,4,7>  <4,5,9> |

**Assignment 15: Classes and Objects**

**Problem description:**

1. Create an Employee class with following attributes:

|  |
| --- |
| **Employee** |
| - empID : int  - empName : String  - empDesig: String  - empDept: String |
| +Employee()  +getEmpID() : int  +getEmpName() : String  +getEmpDesig() : String  +getEmpDept() : String  +setEmpID(int empNo) : void  +setEmpName(String empName) : void  +setEmpDesig(String empDesig) : void  +setEmpDept(String empDept) : void |

Write a java program, which creates an instance of employee class and sets the values for all the attributes.

* While setting value for empName, setEmpName() method should check for NullPointer exception and handle it accordingly.
* While setting value for empDesig, the designation must have any of the following values: developer, tester, Lead or manager. If none of these values is matching then setter method should throw user defined exception 'Invalid designation'.
* While setting value for empDept, the department must have any of the following values: C2, TTG, ITES or PES. If none of these values is matching then setter method should throw user defined exception 'Invalid Dept'.

**Assignment 16: String Compression**

**Problem description:**

Write a program which takes a strings from user and removes all characters in that string except alphabets and then sorts it in lexicographic order.

(The case of the alphabets must be preserved).

**Input:**

Any string.

**Output:**

String consisting of only alphabets sorted lexicographically.

**Sample:**

Input:                    Ax12uZ@s#

Output:                AuxZ

**Assignment 17: String Handling**

**Problem description:**

Given a sentence which consists of alphabets [a-zA-Z], digits[0-9],“,” and “.”.

Complete the provided method to return a string after reversing only the alphabets [a-zA-Z] in a sentence which is passed as an argument to a method.

Note: reversing has to be done word-by-word. Example: if sentence contains “Hello World”, the reversed sentence will be “dlroWolleH”

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 cup of hot coffee costs 8.00, whereas cold coffee costs 45.00. | 1 pucfotoheeffocstsoc 8.00, saerehwdloceeffocstsoc 45.00. |
| It Costs 25000rs for 1 LCD Projector. | tIstsoC 25000sr rof 1 DCL rotcejorP. |
| 8990.33 | 8990.33 |

**Assignment 18: Arrays**

**Problem description:**

Write a console program that reads in a list of integers from user, one per line, until a sentinel value of 0 (which you should be able to change easily to some other value). When the sentinel is read, your program should return array which contains smallest and largest numbers from input list as illustrated in this sample run.



Your program should handle the following special cases:

1. If the user enters only one value before the sentinel, the program should report that value as both the largest and smallest.
2. If the user enters the sentinel on the very first input line, then no values have been entered, and your program should display a message to that effect.

**Input:**

Input is series of numbers

**Output:**

Output is an array, which contains smallest and largest of numbers

|  |  |
| --- | --- |
| **Input** | **Output** |
| 11,17,42,9,-3,35,0 | Array={3,42} |
| 1,0 | Array={1,1} |
| 0 | null |
| 2,50,30,7,8,10,100,3,7,16,0 | Array={1,100} |