|  |  |
| --- | --- |
| Simple | **2017** |

**Capabilities covered:**

* Use Java to implement business layer of your application

**1. Matrix Arrangement:**

Write a function to return an array where the midst position contains the smallest value followed by the next smallest value on the right of the midst position, followed by the next smallest value to the left of the midst position and the rest of numbers continue in this format

Function signature:

public static int [ ] arrangeElements(int[ ] [] inputArray) {

// write the code here

}

|  |  |  |
| --- | --- | --- |
| UTC | Sample Input | Sample Output |
| 01 | |  |  |  | | --- | --- | --- | | 4 | 2 | 13 | | 3 | 8 | 5 | | 9 | 6 | 17 | | {17,9,6,4,2,3,5,8,13} |
| 02 | |  |  |  | | --- | --- | --- | | 4 | 1 | 3 | | 3 | 8 | 5 | | 4 | 16 | 17 | | {17,8,4,3,1,3,4,5,16} |

2. **Consecutive Characters**

Complete the method to print the consecutive characters and the number of times appearing in a String.

public static void printConsecutiveCharacters(String input) {

}

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

**3. Outstanding Persons**

**Time to complete: 45 Minutes**

Outstanding: Distinguished from others in excellence.

We judge a person if he is outstanding or not by the achievement in his/her profession.

3.1.1 Objective:

Understand object oriented paradigm: Constructor chaining, overriding methods, type casting using Run-Time Type Identification and polymorphism.

3.1.2 Problem Statement:

Create classes with generalization-specialization [Inheritance] relationship as shown in UML 

Rules to decide if a person is outstanding:

* Professor is outstanding if he has published more than 4 books
* Student is outstanding if his percentage is greater than or equal to 85.

Using Run-Time Type Identification print Professor and Student details:

The print ( ) method of Professor displays the name and books published by professor.

The display( ) method of Student displays the name and percentage of student.

Details:

The application has to store 5 persons in a single collection of array type

(Person[ ]) , this can be a combination of students and professors, and the application has to display the complete information about the person (name , percentage/books published) only if the person is out-standing.

4.  **Fibonacci Series:**

In [mathematics](http://en.wikipedia.org/wiki/Mathematics), the **Fibonacci numbers** are the numbers in the following [integer sequence](http://en.wikipedia.org/wiki/Integer_sequence):

0,1,1,2,3,5,8,13,21,34,..

By definition, the first two Fibonacci numbers are 0 and 1, and each subsequent number is the sum of the previous two.

In mathematical terms, the sequence *Fn* of Fibonacci numbers is defined by the [recurrence relation](http://en.wikipedia.org/wiki/Recurrence_relation)

Fn = Fn-1 + Fn-2; with inital valuesF0 = 0 and F1 = 1

* 1. **Objective:**

Understand how to write a recursive function.

* 1. **Problem statement:**

Write a method fibonacci that takes some integer nas a parameter and returns the nth Fibonacci number, where we think of the first 1 as the first Fibonacci number. Thus, an invocation of fibonacci(6) should return 8, and in invocation of fibonacci(10) should return 55

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Nth Fibonacci | 0 | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |

5**. Triplets**

Triplets are a set of three similar things.

Complete the function to print all the triplets <A, B, C> such that A+B = C

public static void printTriplets(int[ ] data) {

}

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