

ADVANCE PROGRAMMING (CSE201)

MID SEM EXAM

Monsoon 2022

QUESTION-1: You have to create 3 classes, Professor, Student and Course. Professor has 2 attributes, a name (String), and years of experience (int). Student class also has 2 attributes: student name (String), and rollno (int). The Course class also defines 2 attributes, a Professor object, and an array of 2 Student objects. The Course class should show Association relationship with Professor class, and Composition relationship with Student class.

In the Main class, take 3 Strings and 3 integers as user input (1 pair for Professor, and 2 pairs for Students). Use these values to initialize objects of Professor and Students. Also instantiate 1 object of Course class in the main method. Course class should define 2 methods: dispStudents, and dispProf. These methods use System.out.println to print Professor and Student objects such that their attributes are displayed. These methods should be called in the main method using the object of Course class that has been created.

Note: Just in case this is not clear, you are creating a total of 4 objects: 2 objects of Student, and 1 object each of Professor and Course. **(40% Weightage)**

TEST CASE:

```
Enter the professor's name: Einstein
Enter student 1's name: Bohr
Enter student 2's name: Galileo
Enter the professor's years of experience: 7
Enter student 1's roll number: 224
Enter student 2's roll number: 519
Student Name: Bohr, Roll Number: 224
Student Name: Galileo, Roll Number: 519
Teacher Name: Einstein, Experience: 7
```

QUESTION-2: You have to make use of OOPS concepts taught in lectures to design a system which supports Polynomial operations and the details are as mentioned below

First let's recall what a polynomial is?

A polynomial is defined as an expression which is composed of constants, variables, powers and indeterminates that are combined by means of addition, subtraction, multiplication.

An example of a polynomial with a single variable is given by the following summation:

$\sum_{k=0}^n a_k x^k$ where x is the variable and a_k 's $\forall k \in \{0, 1, \dots, n\}$ are the coefficients of the polynomial.

Polynomials can be of two types -

1. **Polynomial<Integer>** (all coefficients are integers)
2. **Polynomial<Double>** (all coefficients are double values)

Note: you have to create a generic **Polynomial** class such that we can define two objects of type - **Polynomial<Integer>** and **Polynomial<Double>**.

Functionalities common to all types of polynomials are described below -

1. **take input** - You will be given the degree(non-negative) of the polynomial and coefficients of the polynomials which are supposed to be taken as input. An example of the input is as follows:

```
n = 4 //degree
1 0 2 0 3 //coefficients
```

It represents the polynomial : $1 + 0.x + 2.x^2 + 0.x^3 + 3.x^4$

NOTE:- handle the exception where the user inputs the degree as a negative number.

2. **print** - you have to write a function to print polynomial's coefficients in the format
-
 $a_0, a_1, a_2, \dots, a_n$ (a_i represent coefficients.)

3. Now the other important thing to notice is that the coefficients can be of different types ex: Double, Integer etc. For the sake of simplicity we can consider that the coefficients can be of 2 types: either Double or Integer.
4. Below is the detail of all functionalities that Polynomial should have.

Function	Description
setCoefficients, getCoefficients	These are setters/getters of coefficients
addPolynomial, subtractPolynomial	Add / Subtract two polynomials with coefficients having the same data type
getPolynomialGradient	Compute the derivative of the polynomial
printPolynomial	Prints the coefficients of polynomials.
takeInput	Takes the polynomial as input.

NOTE:- you have to make use of Generic Programming, Error Handling, Encapsulation and any other OOPS concepts. **(60% Weightage)**

Test Case for checking working of the Polynomial Class

Test Case 1:

Mention the type of polynomial:

Double

Mention the operation to perform:

Add

Enter the first Polynomial:

4

1.0 1.0 1.0 1.0 1.0

Enter the first Polynomial:

4

1.0 1.0 1.0 1.0 1.0

Result: 2.0 2.0 2.0 2.0 2.0 //print only the coefficients here.

Test Case 2:

Mention the type of polynomial:

Integer

Mention the operation to perform:

getPolynomialGradient

Enter the Polynomial:

4

1 0 0 1 1

Enter the Number to compute the gradient at:

5

Result: 586