American University of Armenia
CS 120 Intro to OOP
Spring 2019

Homework Assignment 7, Part 2

1. **(10 points)** Write a C++ program to calculate the value of a polynomial. Note that the polynomial

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x^1 + a_0$$

can easily be represented as the array $a_0, a_1, a_2, \ldots, a_{n-1}, a_n$ of coefficients. Your program should first input the number n and the coefficients $a_0, a_1, \ldots, a_n$. Then, for each value x inputted by the user, your program should output the corresponding value $f(x)$. The execution of the program should end when there is no more input from the user.

Make sure your program is well-structured, has clearly defined functions with intuitive interfaces. Note that you can rely on partially filled arrays if you don't know how to use dynamic arrays in C++.

2. **(7 points)** Write a C++ program to help a user choose the cheapest taxi service for a journey. The user should first input data about three taxi services, for each using the format:

```
x NONE
```

or

```
x TIP y
```

where `x` is the price per kilometer and `y` is the minimum tip for that taxi service. Then the user inputs the distance they want to travel. After this, the program should output the cheapest fare for that journey and the number of the taxi service offering that fare.

Your program should contain a `calculateFare` function with three parameters for the price per kilometer, the distance and the tip. Note that the tip should have a default argument.

For example, on the input

```
150 NONE
120 TIP 100
100 TIP 350
11
```

your program should output

```
1420 2
```

since taxi service number 2 offers the cheapest fare of 1420 for the journey.

3. **(8 points)** Write a C++ program that inputs the radius of a circle and outputs its circumference and its area. Your program should consist of:

- the `struct Circle`, defined with corresponding member data and member functions;
- the `main` function that illustrates the use of the `Circle` structure.

4. **(20 points)** Develop class `Polynomial`. The internal representation of a `Polynomial` is an array of coefficients. Note that you may need additional data depending on how you implement the array. Develop a complete class containing proper constructor and destructor functions, a `calculateValue` function, as well as accessor and mutator functions. The class should also provide the following overloaded operator capabilities:

- Overload the addition operator (`+`) to add two `Polynomial`s.
- Overload the subtraction operator (`-`) to subtract two `Polynomial`s.
- Overload the assignment operator to assign one `Polynomial` to another.
- Overload the multiplication operator (`*`) to multiply two `Polynomial`s.
- Overload the addition assignment operator (`+=`), subtraction assignment operator (`-=`), and multiplication assignment operator (`*=`).

5. **(5 points)** Write a second C++ program to calculate the value of a polynomial, now relying on the class `Polynomial` from the previous task.