

Advanced Programming

Homework Assignment 7

Polymorphism

General guidelines:

- a. Maximize readability and ensure indentation.
- b. Do exactly as required in the questions.
- c. Every class in every question must be submitted in two separate files – a header file (.h) and an implementation file (.cpp).
- d. Add functions and helper methods as necessary to ensure readability.
- e. Submission must be done according to the submission guidelines – a document published in the course website – read them carefully!
- f. Use relevant names.
- g. Use comments to document your functions and complex code parts. **Add an output example at the end of your file!**
- h. Individual work and submission – paired submission is not allowed.

Important note: Unless otherwise specified, every homework has no more than one week for submission.

Open submission boxes past the deadline are not permission for late submission.

Question 1 (and only):

This exercise continues the implementation of a grant managing system for a higher education institution that we started designing in homework 6. Therefore, the descriptions of the classes Student, BA, MA and PHD are based on the requirements described in homework 6.

Modify the base class **Student** to make it abstract:

- a. The fields will remain with no changes:
 - id – student id
 - firstName, lastName – the first name and the surname
 - numberOfCourses - the number of courses the student is studying
- b. Add and modify the methods as follows:
 - constructor – initializes the fields. Add default values for all the parameters so that numbers will default to 0 and strings will default to an empty string (“”).
 - copy constructor
 - virtual destructor
 - **input()** – change to be pure virtual. The implementation is to stay as it was in homework 6.
 - **print()** – change to be pure virtual. The implementation is to stay as it was in homework 6.
- c. Add the following pure virtual methods without implementation:
 - string **studType()** const – a helper method with protected access.
 - bool **milga()** const – a pure virtual method overridden by the methods described in homework 6 for the classes **BA**, **MA** and **PHD**.

The classes **BA**, **MA** and **PHD** are to remain the same as in homework 6 except for the following changes:

- a. Override the method **studType** so that it returns the string “BA”, “MA” or “PHD” accordingly.
- b. Make sure all the methods that are overriding the pure virtual methods **input**, **print** and **milga** do so correctly as expected in the respective class.

Declare and implement a global function **addStudent**:

- a. The function header is: `Student* addStudent()`.
- b. The object of the function is to allow the user to choose the type of student to input (1 for BA, 2 for MA and 3 for PHD) and to return the address of a new dynamic allocation of the correct type.
- c. After the dynamic allocation the method must call the input method so that the program will input student data the matches the student type.
- d. If the user chooses an invalid choice the function must throw an exception.

Implement a main program that tests your code:

- a. Declare a pointer that can contain the address of an array filled with the different types of students.
- b. Input the size of the array and dynamically allocate it.
- c. Use the function **addStudent** to generate student instances and populate the array. If an exception is thrown, print the error message and repeat the input of the same element.
- d. Iterate over the students and print the details of those students who are eligible to receive a grant.
- e. Free all the memory that requires freeing.

Below is an execution example that demonstrates the expected behavior and output of the program. Make sure your outputs match those in the example.

Execution example:

```
Enter the number of students: 3

Student 1 of 3
Enter degree: 1 for BA, 2 for MA, or 3 for PhD
1
enter id, first name, last name, number of courses
111 Reuven Reuveni 3
and enter a list of student grades
90 95 100

Student 2 of 3
Enter degree: 1 for BA, 2 for MA, or 3 for PhD
5
Exception: no such degree

Student 2 of 3
Enter degree: 1 for BA, 2 for MA, or 3 for PhD
2
enter id, first name, last name, number of courses
222 Shimon Shimoni 7
and enter a list of student grades
88 89 90 91 92 93 94
enter 1 if the student does research and 0 if not
1

Student 3 of 3
Enter degree: 1 for BA, 2 for MA, or 3 for PhD
3
enter id, first name, last name, number of courses
333 Levi Levi 2
enter number of research hours:
26

Students entitled for milga are:

MA student
ID: 222
Name: Shimon Shimoni
Grades: 88 89 90 91 92 93 94
research: YES

PHD student
ID: 333
Name: Levi Levi
Number of research hours: 26
```

Good Luck!