Tartalomjegyzék

Exercises

Open a new project for each exercise or a new file if you're in the command line.

From this point onward use .cpp file extensions!

String extension

Let's extend the last practial's **String** class with the **operator=** and **operator==** methods. The first gives a new value to an existing String. The second compares two Strings.

Example:

```
String s1 = String("batman");
String s2 = String("catman");
cout << (s1 == s2) << endl;
s1 = s2;
cout << (s1 == s2) << endl;
```

Don't forget that when writing **operator=** you have to be careful with the following:

```
String s1 = String("batman");
s1 = s1;
cout << s1 << endl;
```

You can also write methods so that the above two work with C strings as well:

```
String s1 = String("batman");
String s2 = String("catman");
cout << (s1 == "batman") << endl;
s1 = "catman";
cout << (s1 == s2) << endl;
```

Grades

Write a **Grades** class that stores grades for a given subject. It should store names (or neptun codes) and points, which are integers between 0 and 100.

- For every student we should store their name and their points (1 String, 1 integer).
- We can assume that there won't be more than 100 students.
- When we make a new **Grades** object, it should be empty.
- Make a method that appends a new student (their name and points).
- Make a method that returns how many students we're storing currently.
- Create a method that calculates and returns the average of the stored points (as a float).

- With **operator[]** we should be able to retreive the i. student, when i is an integer (the order doesn't actually matter). It should return the name of the student as a String.
- On the other hand if we supply a String to the **operator[]** it should retreive that given student's points and return it as an integer.

Example:

```
Grades grades;
grades.add("Andras", 56);
grades.add("Aladar", 22);
grades.add("Anita", 71);
grades.add("Andrea", 34);
grades.add("Aniko", 64);
for(int i = 0; i < grades.length(); i++) {
  cout << grades[i] << " : " << grades[grades[i]] << endl;
}
cout << "avg: " << grades.avg() << endl;
```

Sorted grades

Modify the above class so that when we retreive a student with **operator[]** and an integer it should retreive the students ordered by their points in descending order. (In other words grades[0] should give the name of the student with the most points.)

- Write a median method that returns the median of the points.
- Make it so a Grades object can be printed through cout. It should print the students' name and points line by line ordered by their points in descending order.
- Extend the class with a new data member: minimum. It stores how many points are required to pass the class. Make a new constructor through which this can be set. By default it should be set to 40.
- Write a **passed** method that returns the names of the students with points above the minimum. Return it as an array of Strings.
- Modify the method responsible for the cout printing so that it prints a line between the students that passed and those who failed. (The line can be for example: "http://wiki.math.bme.hu----------"http://wiki.math.bme.hu.)