

Important message on plagiarism

The single most important point for you to realize before the beginning of your studies at ShanghaiTech is the meaning of “plagiarism”:

Plagiarism is the practice of taking someone else's work or ideas and passing them off as one's own. It is the misrepresentation of the work of another as your own. It is academic theft; a serious infraction of a University honor code, and the latter is your responsibility to uphold. Instances of plagiarism or any other cheating will be reported to the university leadership, and will have serious consequences. Avoiding any form of plagiarism is in your own interest. If you plagiarize and it is unveiled at a later stage only, it will not only reflect badly on the university, but also on your image/career opportunities.

Plagiarism is academic misconduct, and we take it very serious at ShanghaiTech. In the past we have had lots of problems related to plagiarism especially with newly arriving students, so it is important to get this right upfront:

You may...

- ... discuss with your peers about course material.
- ... discuss generally about the programming language, some features, or abstract lines of code. As long as it is not directly related to any homework, but formulated in a general, abstract way, such discussion is acceptable.
- ... share test cases with each other.
- ... help each other with setting up the development environment etc.

You may not ...

- ... read, possess, copy or submit the solution code of anyone else (including people outside this course or university)!
- ... receive direct help from someone else (i.e. a direct communication of some lines of code, no matter if it is visual, verbal, or written)!
- ... give direct help to someone else. Helping one of your peers by letting him read your code or communicating even just part of the solution in written or in verbal form will have equal consequences.
- ... gain access to another one's account, no matter if with or without permission.
- ... give your account access to another student. It is your responsibility to keep your account safe, always log out, and choose a safe password. Do not just share access to your computer with other students without prior lock--out and disabling of automatic login functionality. Do not just leave your computer on without a lock even if it is just for the sake of a 5--minute break.
- ... work in teams. You may meet to discuss generally about the material, but any work on the homework is to be done individually and in privacy. Remember, you may not allow anyone to even just read your source code.

With the Internet, "paste", and "share" are easy operations. Don't think that it is easy to hide and that we will not find you, we have just as easy to use, fully automatic and intelligent tools that will identify any potential cases of plagiarism. And do not think that being the original author will make any difference. Sharing an original solution with others is just as unethical as using someone else's work.

CS100 Homework5

Overview

In this homework, you are required to do some simple Object-oriented programs with C++ programming language.

This homework contains 3 individual programs.

- Percentage of this homework over the whole score: 11%

Submission deadline

2019-11-6 23:59

To encourage debugging locally and not on the OJ output (which is not right for practical programming), the OJ will not be available from the date the homework is released. It will gradually open until the submission deadline passes, and the schedule is:

- Part 1 submission will be available on 10/31 at 17:00.
- Part 2 submission will be available on 11/2 at 17:00.
- Part 3 submission will be available on 11/4 at 17:00.

Problem 1. Data Analyser

Description

In this Problem, we provide a base class named `DataAnalyser`. In this class, we've already implemented the constructor, destructor, and a virtual function called `calculate`, as shown below.

```
class DataAnalyser
{
public:
    DataAnalyser() {};
    virtual ~DataAnalyser() {};
    virtual float calculate(float* data, size_t size)
    {
        std::cout << "ERROR: Try to access virtual function in base
class\n";
        return ERRORCODE;
    };
};
```

Explanations of these functions

- *`DataAnalyser :: DataAnalyser()`*

The constructor of base class

- *`virtual DataAnalyser :: ~DataAnalyser()`*

The destructor of base class

- *`float DataAnalyser :: calculate(float* data, size_t size)`*

A virtual function which takes 2 parameters, the first one, `data`, is an array that need to be calculated, and the second one, `size`, is the size of the array, returns the calculated result.

You are required to implement 5 classes called `MaximumAnalyser`, `MinimumAnalyser`, `AverageAnalyser`, `MedianAnalyser` and `StdDevAnalyser` inherits from the given `DataAnalyser` (i.e. They are child classes of `DataAnalyser`), you need to overload the function `calculate` in these child classes.

- For `MaximumAnalyser`, calling function `calculate` will return the maximum value in `data`.
- For `MinimumAnalyser`, calling function `calculate` will return the minimum value in `data`.
- For `AverageAnalyser`, calling function `calculate` will return the average(mean) of `data`.

Here is the formula to calculate the average of `data`:

Let $data = [x_1, x_2, \dots x_n]$

Then,

$$\text{Avg}(\text{data}) = \frac{\sum_{i=1}^n (x_i)}{n}$$

- For **MedianAnalyser**, calling function **calculate** will return the median of **data**.
- For **StdDevAnalyser**, calling function **calculate** will return the standard deviation of **data**.

Here is the formula to calculate the standard deviation of **data**:

Let $\text{data} = [x_1, x_2, \dots, x_n]$

Then,

$$\text{StdDev}(\text{data}) = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{Avg}(\text{data}))^2}{n}}$$

where $\text{Avg}(\text{data})$ is the average number of data.

- Note that, if your implementation is correct, no "ERROR" information will be printed when running your program.

Requirements

- You are required to implement 5 different child classes inherits from **DataAnalyser**, each child class can calculate one property of given **data**. You need to overload **calculate** function, to give different performance in child classes. You CANNOT modify the name or arguments list of the **calculate** function or the name of these classes, or compile errors will occur.
- If you want to add your own functions and classes, feel free to add them.
- For more details, see the template code given to you.

Debugging & Testing

- To make you debug easier, we provide a some useful codes for you, you can modify it as you want. The following codes should be added to your **main** function, so, **DO NOT submit them to OJ**.

```
/* You can change these numbers to whatever you want*/
float arr[] = {0.3, -100, 1, 2, 5, 6, 9, 12, 2};
DataAnalyser* foo = new MinimumAnalyser(); /* it can be any of the
required 5 child classes */
std::cout << foo->calculate(arr, 9) << std::endl; /* should be -100 in
this case */
delete foo;
```

Submission

- When submitting it to online judge, you need to submit the base class and 5 child classes you write, If you add additional includes, classes, functions, variables or definitions, please also submit them to OJ.
- **DO NOT submit the `main()` function**, or compile errors will occur.

Problem 2. Structured outputting in console

Description

Assume you are one of TAs of CS999. Oneday, *Homework??* was released. It was a huge project which contains m different student informations, like name, email, or scores for each problem, etc. After the deadline, you need to collect all n students' information and send the report to professor.

- The original report you downloaded from OJ contains n lines, matching to n students.
- For each line, it contains m numbers separated by a single whitespace (i.e. " "), representing someone's informations (i.e. "Madoka Madoka@shanghaitech.edu.cn 99 100 81").
- For instance, $n = 3$ and $m = 4$ represents there are 3 students and 4 informations per student, a sample score sheet is shown below.

```
Umi 20 aa 43
Honoka 43253 65789 87912
Kotori 1.7 foo 44
```

But the professor of CS999 is extremely strict with formats of reports, here are the requirements.

- The first line of the report should starts with a "/", ends with a "\", interiors are filled with "-".
(e.g. "/-----\"")
- The last line of the report should start with a "\", ends with a "/", interiors are filled with "-".
(e.g. "\"-----/")
- For middle lines, it can be (a). "line contains numbers" or (b). "separate-line", they are all start and end with a bar-separator ("|"), and occurs alternately (i.e. abababa).
 - **line contains information:** these lines only contain bar-separators ("|"), informations (e.g. 100, GeZiWang, 1.233, OOP), and whitespaces (" "). There are **at least one whitespaces** between each "information" and "bar-separators", and numbers are **left-aligned**.
 - (e.g. "| 100 | 200 | 300 |")
 - **separate-line:** these lines only contain bar-separators ("|") and minus signs ("-"). The "|"s should be aligned with other lines.
 - (e.g. "|-----|-----|-----|")
- **Note: All bar-separators should be aligned, all informations are left-aligned**
- **Note: All lines you output should have the same length (same number of characters).**
- The students should be **sorted** alphabetically by the i^{th} column (starts from 0) of their information (e.g. If sorting by column 0, Honoka > Kotori > Umi in case shown above).
 - `std::string` provides you an **operator <**, so you will not need to write your own string comparison.

- If you want to swap student **a** and student **b**, you should swap the **whole row**, not only the given column, otherwise the informations will be messed up.

e.g. if the data is

```
b 12
a 123
```

you should sort it to this (by column 0):

```
a 123
b 12
```

- `std::sort` may be very useful for this requirement.
- Hint: you can store all the informations as strings, instead of using multiple types.
- For instance, the following report is the structured report of report shown above (sorting by col 0).

```
/-----\
| Honoka | 43253 | 65789 | 87912 |
|-----|-----|-----|-----|
| Kotori | 1.7   | foo   | 44    |
|-----|-----|-----|-----|
| Umi    | 20    | aa    | 43    |
\-----/
```

So, you want to write a class called **ReportParser** to structure the report, and give the structured one. We also provide a template for you (as shown).

```
class ReportParser
{
public:
    // The constructor
    ReportParser(int numStudents, int numInfos);
    // The destructor
    ~ReportParser();
    // read & write functions
    void readReport();
    void writeStructuredReport(int sortOption);
    // Add your own functions and variables here
private:
    // Add your own functions and variables here
};
```

Some functions are provided for you.

- *ReportParser :: ReportParser(int numStudents, int numProblems)*

The constructor of ReportParser, **numStudents** is the number of students, **numInfos** is the number of informations.

- *ReportParser :: ~ReportParser()*

The destructor of ReportParser

- *void ReportParser :: readReport()*

Read the input report from the console (stdin)

- *void ReportParser :: writeStructuredReport(int sortOption)*

Write the structured report to the console (stdout)

Requirements

- The input contains **n+1** lines, the first line contains 3 numbers **m**, **n** and **sortOption**, represents the info number per line, total student number, and the result should be sorted by which column (sortOption is an integer in $[0, m - 1]$). For the following **n** lines, each line contains **m** informations seperated by " ", representing **m** informations for this student.
- You need to implement the **ReportParser** class, read the input report and structured it into the required format.
- Be careful about the requirements of **structured report** shown above, especially the **bar-separators** alignment, **infos** alignment and whitespaces.
- Your output is a structured report.
- **main()** function is also provided to you in the template code, you don't need to modify it.

Submission

- When submitting it to online judge, you should submit all your code except **main()**. If you add additional includes, classes, functions, variables or definitions, please also submit them to OJ.
- **DO NOT submit the main() function**, or compile errors will occur.

Problem 3. Lookup Table

Description

In this problem, you are required to implement a simple lookup table. This is, users has some variable x , and a function $f(x)$, they want to know what's the result of $f(x)$. A simple idea is, we pre-calculate $f(x)$ for all x s, but that's not possible for a continuous function. But we can use sampling to discrete it, and use a vector to store these discrete (x, y) points.

i.e. values of $f(x) = x^2$ which start from 0, ends at 5 and increments by 1 can be stored as

x		0		1		2		3		4		5
--		---		---		---		---		---		---
y		0		1		4		9		16		25

In the beginning (class constructor), we will give the start position x_0 , end position x_n , and an increment dx .

You need to calculate the $f(x)$ value ranges from $x = x_0$ to $x = x_n$ with an increment dx

Then, users may input some arguments x that they want to find the value of $f(x)$.

For example, for $f(x) = x^2$, if some user input a number 2, you will output a number 4 because $2^2 = 4$. Here are some notes you need to pay attention to:

- If user wants to lookup some x out of the range $[x_0, x_n]$, for example, if someone try to find 10^2 , which is not in the input range shown above, your program should call an error handling function **InputOutOfRangeError** that provides to you, and just returns 0 for error.
- If someone tries to lookup a value in the input range, but there are no actually x values matching, you should use a linear approximation to give the user an approximate value.

That is, if someone try to find $f(x_{user})$, but there are no x_{user} stored in your database. The two x s which have least difference with x_{user} is x_k and x_{k+1} , then the value of $f(x_{user})$ is given by the following formula:

$$f(x_{user}) = \frac{(x_{user} - x_k) * y_{k+1} + (x_{k+1} - x_{user}) * y_k}{x_{k+1} - x_k}$$

For example, if someone tries to lookup 2.3 in the database shown above, then, we can get 2.3^2 by the following formula:

$$f(2.3) = \frac{(2.3 - 2) * f(3) + (3 - 2.3) * f(2)}{3 - 2}$$

which equals to 5.5, respectively.

We provide a pure virtual base class for you to do this job, you need to complete 3 classes inherits from this base class. The base class is like this:

```
class LookupTable
{
public:
    // constructor, start: start position; end: end position; increment:
    the increment "dx"
    // You should calculate the value in range [start, end], with
    increment "increment"
    LookupTable(double start, double end, double increment);
    // virtual destructor
    virtual ~LookupTable() = 0;
    // get the value f(x) of the given x
    virtual double getValue(double x) = 0;
};
```

- *double LookupTable :: getValue(double x)*

Get the value of $f(x)$, where x is the given argument **double x**.

You need to return the $f(x)$ value by searching in your database.

Your job is to implement 3 child classes (inherit from **LookupTable**) named **SquareLookupTable**, **SinLookupTable**, and **LogLookupTable**.

- In their constructors, you should calculate the values of $f(x)$ from **start** to **end**, with increment **increment** using functions from `<math.h>`, and store it in a vector.
- Calling the function `getValue(a)` in **SquareLookupTable**, **SinLookupTable**, and **LogLookupTable** will return a^2 , $\sin(a)$, $\log(a)$.

// $\log(a)$ is the same as $\ln(a)$

Requirements

- In this problem, you are required to implement a lookup table.
- Your job is to complete 3 child classes shown above.

Submission

- When submitting it to online judge, you need to submit your **LookupTable** class and all child classes. If you add additional includes, classes, functions, variables or definitions, please also submit them to OJ.
- **DO NOT submit the `main()` function**, or compile errors will occur.