

Assignments C/C++ programming (2008/2009)

General requirements

Compiler: The programs you submit should compile without errors by GNU C/C++ compiler g++ and run on kits.few.vu.nl server.

Style: You should comply with the following coding style guidelines:

- use C++ features and NOT C features (eg. cout instead of printf).
- use functions! Make your code modular.
- document your code: comment on every function's arguments and return value. Give a brief description of its important features.
- use indentation.
- use meaningful variable and function names.

Deadlines:

- assignment 1 : Monday, 09-02-2009, 23:59
- assignment 2 : Monday, 16-02-2009, 23:59
- assignment 3 : Monday, 23-02-2009, 23:59
- assignment 4 : Monday, 2-03-2009, 23:59

Submit and feedback rules

- The assignments are individual. Please write in comments at the beginning of each program your name, student number and your studygroup.
- The programs have to be submitted by e-mail to your assistant before the specified deadlines. Please include the word 'assignment' and the assignment number in the subject of your mail. Send the source code only, not the executables.
- Feedback is guaranteed in 24h. Expect no feedback on Saturday and Sunday.
- After the first submission you get from your assistant a list with things that have to be corrected. Please follow this list !!
- Each student can submit max 3 times a solution. The first solution has to be a working program. If by next deadline the third solution is still not satisfying, the student is declared failed.

Integrity and ethics. You can discuss about the same assignment but two identical solutions will NOT be accepted. Remember that representing someone else's work as one's own is grounds for failing the course.

Assignment 1. Warm-up (due at midnight Monday, 9 February, 2009)

Write small C++ pieces of code according to each of the following scenarios'.

Write comments for each relevant C++ statement so that the assistant can see what you do and where.

You have to deliver one program for each scenario.

You also have to deliver a results sheet where you write down the results printed by your programs and where asked also an explanation or a drawing.

- a) Here you can write everything in a `main()` function. Suppose `fred` is a double variable. Assign `fred` a value. Display the value of `fred`. Declare a pointer that points to `fred`. Display the value of `fred` using this pointer. Display the address of `fred` in decimal and hexadecimal format, with and without pointer use. Change the value of `fred` by using the pointer. Display the value of `fred` again with and without pointer use. Write down the results.
 - b) Create an array of 10 integers and assign to each element a different value. Display all the elements and their addresses (in decimal and hexadecimal) in memory using the `&` operator. Declare a pointer to the array. Display all the elements and their addresses using this pointer and pointer arithmetic. Make a drawing with memory locations, their addresses and their content to illustrate how exactly are the array elements being stored.
 - c) Ask the user to enter a positive integer and create a dynamic array of that many integers. Fill in this array with some values. Write a function `display(int* array_ptr, int n)` that gets as parameter a pointer to an array and an integer `n` and displays the first `n` elements of this array. Use this function to display the elements of the array you created.
 - d) Write a function `int* generate_rand (int length, int MAX_ARRAY)` that gets an integer as input and generates a dynamic array with the same number of elements, fills the array with random values between 0 and `MAX_ARRAY` and returns a pointer to this array.
Write a function `calculate (int* array_ptr, int length, &mean, &sd)` that gets a pointer to an array and its length and returns the mean and standard deviation of its elements.
Use these two functions in a `main()` function where you ask the user for an array length, you generate an array of that length filled with random elements between 0 and 100 and you calculate the mean and standard deviation for the generated values.
Run the program 3 times while increasing the array length (for example, for 100, 1000, 10000 elements) and write down the results. Can you explain the obtained values for the mean?
-

Implementation notes. The function `rand()`, located in `<cstdlib>` header generates random integer numbers between 0 and `RAND_MAX`, a constant defined in `cstdlib`. For example you can generate numbers between 0 and 99 if you use `rand() % 100`. Read more in any C++ online reference.

For a set of n values, x_1, x_2, \dots, x_n , $n > 1$, the average or mean is given by

$$\bar{x} = \frac{\sum x_i}{n},$$

The standard deviation is given by: $s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{(n-1)}}$

Assignment 2. AEX index (due at midnight Monday 16 February 2009)

The Amsterdam Exchanges Index (AEX) is a weighted index based on the prices of shares in a fixed number of leading Dutch companies. The AEX index is calculated as the value in euros of a basket with shares from these companies. Each company fund participates in the Amsterdam stock exchange with a certain volume of shares. This volume determines with how many shares each company participates in the AEX basket (the relative weight of the company in the AEX index).

The composition of the AEX index and the relative weights for the participating companies are fixed each year. As for March 2009, the AEX index has the following composition (AEX samenstelling), showing for each company fund the number of shares participating in the AEX basket (**AEX2009.txt**) :

Aegon: 181.00
Ahold: 132.00
Akzo Nobel: 30.00
Arcelor Mittal: 121.00
ASML Holding: 48.00
Bam Group: 12.00
Corio: 5.50
DSM: 15.00
Fortis: 242.00
Fugro: 6.50
Heineken: 27.00
ING: 247.00
KPN: 203.00
Philips: 127.00
Randstad: 16.00
Reed Elsevier: 73.00
Royal Dutch Shell-A: 286.00
SBM Offshore: 16.00
TNT: 41.00
TomTom: 17.00
Unibail: 9.00
Unilever: 187.00
USG People: 8.00
Wereldhave: 2.50
Wolters Kluwer: 34.00

Table 1. The AEX index composition (file AEX2009.txt)

The share price for each fund fluctuates during the day according to the stock transactions, but the closing share price (slotkoers) at the end of each trading day are published in the media.

The AEX index is also recalculated every 15 seconds, but the last calculated value of the day is also quoted in the media (www.euronext.nl). In this period the AEX is around 260.

Table 2 and Table 3 show extracts from the newspapers from 10th and 12th of January 2009, with the share closing values (in euro) for the AEX participating funds.

Aegon: 5.1
Ahold: 8.71
Akzo Nobel: 31.81
Arcelor Mittal: 20.41
ASML Holding: 12.85
Bam Group: 7.6
Corio: 38.00
DSM: 20.06
Fortis: 1.12
Fugro: 23.15
Heineken: 24.22
ING: 7.92
KPN: 10.59
Philips: 14.64
Randstad: 15.94
Reed Elsevier: 9.11
Royal Dutch Shell-A: 20.29
SBM Offshore: 9.99
TNT: 15.83
TomTom: 5.69
Unibail: 114.99
Unilever: 17.74
USG People: 9.56
Wereldhave: 64.81
Wolters Kluwer: 14.3

Table 2. The closing share prices for AEX funds (from Het Parool, Aandelen AEX-index, 10 januari 2009) (file AEX-10jan-2009.txt)

Aegon: 5.27
Ahold: 8.85
Akzo Nobel: 31.42
Arcelor Mittal: 20.14
ASML Holding: 12.61
Bam Group: 7.51
Corio: 38.11
DSM: 19.99
Fortis: 1.16
Fugro: 23.43
Heineken: 23.91
ING: 8.5
KPN: 10.67
Philips: 14.75
Randstad: 16.3
Reed Elsevier: 9.34
Royal Dutch Shell-A: 20.21
SBM Offshore: 9.82
TNT: 15.3
TomTom: 5.73
Unibail: 115.88
Unilever: 17.72
USG People: 9.39
Wereldhave: 64.51
Wolters Kluwer: 14.71

Table 3. The closing share prices for AEX funds (from Het Parool, Aandelen AEX-index, 12 januari 2009) (file AEX-12jan-2009.txt)

Write a C++ program to analyze the Amsterdam stock exchange market situation on 12th January 2009.

1. The program has to read data from 3 files: the AEX composition **AEX2009.txt**, the share values from 10 January, **AEX-10jan-2009.txt** and the share values from 12 January, **AEX-12jan-2009.txt**.

2 The program has to calculate the AEX index for 12 January, indicating its relative change compared with 10 January (for example AEX= 260 + 0.64 %)

3. The program has to write for each company the absolute and relative change of its shares price.

4. The program has to determine the top riser and the top faller company (grootste stijger, grootste daler in relatieve verschil).

Notes on implementation:

It is not necessary to use linked lists.

You have to solve this assignment in a generic way. For example, do not hardwire the number of companies because it can happen during the year that the number of participating companies changes.

If a quantity is today x and tomorrow y , its absolute change is $(y-x)$. The relative change is $(y-x)/x$ or $(y-x)*100/x$ in [%].

Steps to calculate the AEX index:

1. Multiply for each company fund the share closing price (in euros) with the number of shares in the AEX basket. For example: ABN-AMRO 21.94 x 190.00
2. Add all the products.
3. Divide the result by 100. This is the AEX index.

An example of AEX calculation you can find on:

<http://www.behr.nl/Beurs/aex.html>

Assignment 3. Company (due at midnight Monday, February 23, 2009)

The relevant information about a company employees is stored in a text file **employees.txt** (see Table 1). Each line in the file stores personal information about one employee, specified by its personal ID, name, function and annual salary, separated by a semicolon.

```
13540; Johnson, Alan;Clerk;25100
10003; Lockers, Daniel;Engineer;53500
25782; Adamson, John;Manager;77500
09991; Newhall, Bob;Programmer;40000
37829; Brown, James;Clerk;28000
23672; Cheng, Linda;Accountant;35500
12998; Lipmann, Charles;Programmer;45500
67413; Watt, James;Electrician;30000
90786; Jones, Mary;Manager;70500
12343; Lockers, Dennis;Programmer;41000
87962; Green, Sylvester;Clerk;30000
04515; Lubbers, Pieter;Scientist;60000
```

Table. 1. The contents of the input file **employees.txt**

Write a program to manage personal employee information for this company by using a linked list.

1. The program has to read the employees file and add employee information to the list. The information has to be inserted in the list so that the list remains always sorted alphabetically by the employee name.
2. The program has to write the sorted list to the standard output and to a file.
3. The program has to calculate and print the number of employees and their average salary.
4. The program asks the user for the personal ID of an employee that left the company, X. The program looks for this employee and if found, deletes his record from the list and prints again the list. If the ID is not found, the program returns a message "Not found" and does nothing.

Notes on implementation:

You have to solve this assignment by using classes and a singly linked list of employees. You may not use any C++ Standard Template Library (STL) classes.

The class List has the following member functions:

- a default constructor
- a destructor
- a function to insert an element in the right position in the sorted list
- a function to delete an element with the personal ID X
- a `friend` function to print the list

You will need the following struct outside the list class for the list nodes:

```
struct node
{
    long personal_ID;
    string name;
    string function;
    long salary;
    node *next;
};
```

You have to create 3 separate files, the interface in `list.h` file, the implementation in `list.cpp` file and the test file in `company.cpp`.

Use **gmake** utility to compile and link your executable program.

You have to submit all 3 files and the **makefile** to the assistants.

Assignment 4. Top 10 Dutch tourist attractions (due at midnight Monday March 2, 2009)

The Dutch tourism office issues each year a report Kern Cijfers Tourisme en recreatie with main facts about holidays spent by foreigners in Netherlands and by Dutch citizens abroad. The file `visitors.txt` contains the main tourist attractions in the Netherlands and the number of visitors for each attraction registered in 2005 (in thousands).

1. Write a program to process the information about the main tourist attractions. The program should display on the screen the names of the top 10 most visited attractions and for each of them the percentage of total visitors.

Anne Frank Huis,	937
Rijksmuseum,	810
Rondvaarten Amsterdam,	2500
Van Gogh Museum,	1338
Dolfinarium,	720
De Keukenhof,	720
Diergaarde Blijdorp,	1500
Noorder Dierenpark Emmen,	1410
De Efteling,	3253
Dierenpark Amesfoort,	685
Atractiepark Slagharen,	1325
Duinrell,	1310
Natura Artis Magistra,	1082
Ouwehands Dierenpark,	825
Burger's Zoo,	1515
Madurodam,	630
Openlucht museum de Zaanse Schans,	880

Table. 1. The main Dutch tourist attractions and their visitors (in thousands) (file `visitors.txt`)

2. Write the results to a file and produce a pie chart with the top 10 attractions and a slice for all the others.

Notes on implementation

You have to use the template class `vector` and the generic `sort` algorithm from the STL.

You have to define a structure for data consisting of one tourist attraction and the number of visitors.

In order to use the sorting algorithm you have to write a function to overload the `<` operator and to accept arguments of structure type.

You also have to overload the operator `<<` to print a structure on the standard output.

For the pie chart you can use Excel or any other software package. Deliver to your assistant the C++ program and the pie chart, while mentioning which tool you used.