

CS201 – Spring 2015-2016 - Sabancı University
Homework #1 – Netflix Internet Usage
Due February 17, Wednesday, 19:00 (Sharp Deadline)

Description

In this homework, you will write a C++ program that performs some basic calculations on user entered data. After the launch of Netflix in Turkey, it is possible to watch lots of movies and TV series via streaming. Streaming high definition videos requires huge amounts of data usage. However, in Turkey there is a limit on personal internet usage in order to provide “fair experience” to every user. The C++ program that you will develop will calculate the total data usage of Netflix in a given month, and the remaining quota for that month.

For our scenario, the total quota (in GBs) on the user’s internet connection will be taken as an input. We will assume that the required data transfer to watch a movie and a single episode of any TV series (both in HD) are fixed values. You may find these values in the following table as MBs. You will need to use them in your calculations.

Average amount of data transfer for a movie	6144
Average amount of data transfer for a single episode	2520

Then, using the inputs received and fixed values you will calculate the total data usage of Netflix (in MBs) and the remaining quota (in GBs).

Please note that, you need to make conversions between megabytes (MBs) and gigabytes (GBs) in order to use inputs in a correct way, and display the outputs in the expected format.

All program flow may be computed in the *main* function, that is you are not expected to write any user-defined functions, but of course you can write some, if you prefer to do so.

Programming Language/Environment

You should prepare (or at least test) your program using MS Visual Studio 2012 C++. We will use the standard C++ compiler and libraries of the above mentioned platform while testing your homework.

Input and Output

There are three **inputs** in this program:

- Total quota of the user’s internet usage in **GBs** (of type `double`),
- Number of movies watched by the user (of type `int`),
- Number of episodes watched by the user (of type `int`),

Two **outputs** are expected in this program:

- Total internet usage of Netflix in **MBs** (of type `double`),
- Remaining quota in **GBs** (of type `double`).

Program Flow

At the beginning, the user will be prompted to enter the previously mentioned three inputs. The order of the inputs should be exactly the same as the way given in the previous section. Then, necessary calculations should be made by your program. You may refer to the following formula for the calculations:

q_t : total quota

q_r : remaining quota

n_m : number of movies

n_e : number of episodes

u_n : total Netflix usage

d_m : required data transfer for a movie

d_e : required data transfer for a single episode

$$u_n = (n_m * d_m) + (n_e * d_e)$$

$$q_r = q_t - (u_n / 1024)$$

After completing calculations, total usage of Netflix and remaining quota should be displayed in this order. Please refer to the Sample Runs section to see the program flow.

VERY IMPORTANT!

Your programs will be compiled, executed and evaluated automatically; therefore you should definitely follow the rules for inputs and outputs. See **Sample Runs** section for some examples.

- **Order of inputs and outputs** must be in the abovementioned format.

The textual messages in the prompts (for the inputs) and in the outputs are your design decisions. However, it is very important for you to follow the order of inputs and outputs explained above. Following these rules is crucial for grading, otherwise our software will not be able to process your outputs and you will lose some grades in the best scenario.

Input Check

You do not need to perform any kind of input checks; you may assume that the user enters positive values correctly for inputs and you may also assume that the total Netflix usage will not exceed the total quota.

IMPORTANT!

If your code does not compile, you will get **zero**. Please be careful about this and double check your code before submission.

Sample Runs

Below, we provide some sample runs of the program that you will develop. The italic and bold phrases are inputs taken from the user. You should follow the input and output order in these examples.

Sample Run 1

This program bla bla bla...
Please enter your internet usage limit in GBs: **150**
Please enter the number of movies you watched in this month: **8**
Please enter the number of episodes you watched in this month: **22**
Total Netflix usage in MBs: **104592**
Remaining quota in GBs: **47.8594**

Sample Run 2

This program bla bla bla...
Please enter your internet usage limit in GBs: **100**
Please enter the number of movies you watched in this month: **3**
Please enter the number of episodes you watched in this month: **19**
Total Netflix usage in MBs: **66312**
Remaining quota in GBs: **35.2422**

Sample Run 3

This program bla bla bla...
Please enter your internet usage limit in GBs: **100**
Please enter the number of movies you watched in this month: **0**
Please enter the number of episodes you watched in this month: **30**
Total Netflix usage in MBs: **75600**
Remaining quota in GBs: **26.1719**

Sample Run 4

This program bla bla bla...
Please enter your internet usage limit in GBs: **200**
Please enter the number of movies you watched in this month: **26**
Please enter the number of episodes you watched in this month: **0**
Total Netflix usage in MBs: **159744**
Remaining quota in GBs: **44**

Sample Run 5

This program bla bla bla...
Please enter your internet usage limit in GBs: **187.5**
Please enter the number of movies you watched in this month: **5**
Please enter the number of episodes you watched in this month: **64**
Total Netflix usage in MBs: **192000**
Remaining quota in GBs: **0**

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all homeworks, unless otherwise noted.

How to get help?

You may ask questions to [TAs](#) (Teaching Assistants) of CS201. Office hours of TAs are at the class [website](#). Recitations will partially be dedicated to clarify the issues related to homework, so it is to your benefit to attend recitations.

Moreover, in the recitations for the first homework, there will be a demonstration on how to prepare homework for submission and how to submit it to SUCourse. This process is not so straightforward. We have experienced several unsuccessful submissions in the previous years. Therefore, we strongly recommend you attend this demo.

What and Where to Submit

Please see the detailed instructions below/in the next page. The submission steps will get natural/easy for later homeworks.

Grading and Objections

Careful about the semi-automatic grading: Your programs will be graded using a semi-automated system. Therefore you should follow the guidelines about input and output order; moreover you should also use same prompts as given in the Sample Runs. Otherwise semi-automated grading process will fail for your homework, and you may get a zero, or in the best scenario you will lose points.

Grading:

- ☐ Late penalty is 10% off of the full grade and only one late day is allowed.
- ☐ **Having a correct program is necessary, but not sufficient to get the full grade. Comments, indentation, meaningful and understandable identifier names, informative introduction and prompts, and especially proper use of required functions, unnecessarily long program (which is bad) and unnecessary code duplications (which is also bad) will also affect your grade.**
- ☐ Please submit your own work only (even if it is not working). It is really easy to find out “similar” programs!
- ☐ For detailed rules and course policy on plagiarism, please check out http://myweb.sabanciuniv.edu/gulsend/su_current_courses/cs-201-spring-2008/plagiarism/ and keep in mind that

Plagiarism will not be tolerated!

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your homework from the email address provided in the comment section of your announced homework grade or attend the specified objection hour in your grade announcement.

- Check the comment section in the homework tab to see the problem with your homework.
- Download the .zip file you submitted to SUCourse and try to compile it.
- Check the test cases in the announcement and try them with your code.
- Compare your results with the given results in the announcement.

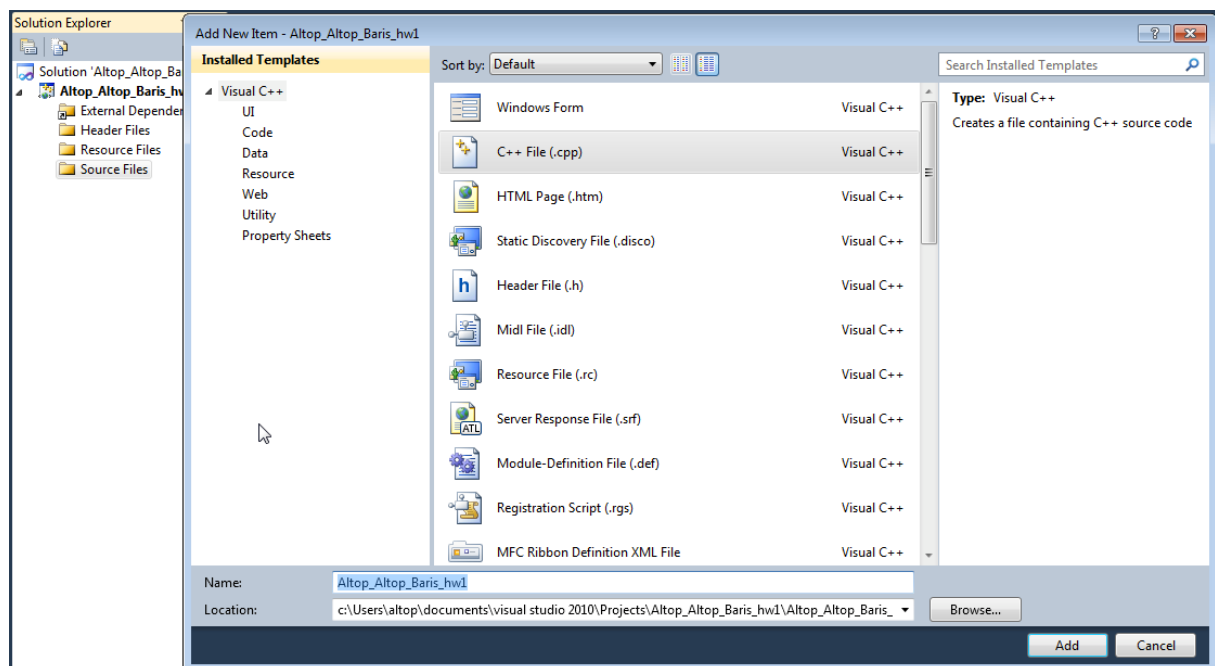
What and where to submit (IMPORTANT)

Submissions guidelines are below. Most parts of the grading process are automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Add your name to the program: It is a good practice to write your name and last name somewhere in the beginning program (as a comment line of course).

Name your submission file:

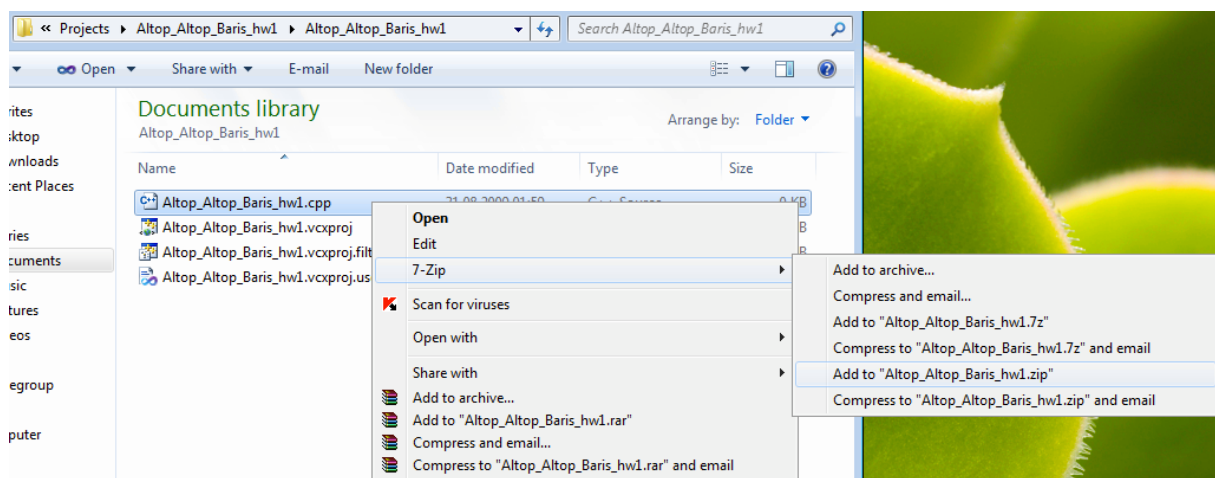
- ☐ Use only English alphabet letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
- ☐ Name your cpp file that contains your program as follows.
“SUCourseUserName_YourLastname_YourName_HWnumber.cpp”



- ❑ Your SUCourse user name is actually your SUNet user name which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsizkodyazaroglu, then the file name must be:

Cago_Ozbugsizkodyazaroglu_Caglayan_hw1.cpp

- ❑ Do not add any other character or phrase to the file name.
- ❑ Make sure that this file is the latest version of your homework program.
- ❑ Compress this cpp file using WINZIP or WINRAR programs. Please use "**zip**" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension.



- ❑ Check that your compressed file opens up correctly and it contains your **cpp** file. You will receive no credits if your compressed zip file does not expand or it does not contain the correct file.
- ❑ The naming convention of the zip file is the same as the cpp file (except the extension of the file of course). The name of the zip file should be as follows.

"SUCourseUserName_YourLastname_YourName_HWnumber.zip"

For example zubzipler_Zipleroglu_Zubeyir_hw1.zip is a valid name, but hw1_hoz_HasanOz.zip, HasanOzHoz.zip are NOT valid names.

Submission:

- ❑ Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).
 - 1) Click on "Assignments" at CS201 SUCourse (not the CS201 web site).
 - 2) Click Homework 1 in the assignments list.
 - 3) Click on "Add Attachments" button.
 - 4) Click on "Browse" button and select the zip file that you generated.
 - 5) Now, you have to see your zip file in the "Items to attach" list.
 - 6) Click on "Continue" button.
 - 7) Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Resubmission:

- ❑ After submission, you will be able to take your homework back and resubmit. In order to resubmit, follow the following steps.
- 1) Click on "Assignments" at CS201 SUCourse.
 - 2) Click Homework 1 in the assignments list.
 - 3) Click on "Re-submit" button.
 - 4) Click on "Add/remove Attachments" button
 - 5) Remove the existing zip file by clicking on "remove" link. This step is very important. If you do not delete the old zip file, we receive both files and the old one may be graded.
 - 6) Click on "Browse" button and select the new zip file that you want to resubmit.
 - 7) Now, you have to see your new zip file in the "Items to attach" list.
 - 8) Click on "Continue" button.
 - 9) Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!

Ece Egemen and Gülşen Demiröz