

# CS201 – Fall 2019-2020 - Sabancı University

## Homework #2: Coffee Quiz

**Due October 16, Wednesday, 22:00 (Sharp Deadline)**

### Introduction

The aim of this homework is to practice on parametric functions and if statements. The use of if statement is necessary in this problem; however, the use of functions is required for good programming style. That means, it is possible to accomplish this homework without using functions, but it is a must to use them. The details about the use of functions in this homework are given later in this document.

### Description

In this homework, you will write a C++ program that will help a barista know what kind of coffee customer would like to drink! Structured as a quiz program, your program will require the users to answer several questions based on ingredients used in making coffee, and notifies what kind of coffee has been prepared by the barista. After displaying a notification about this information, the program will close.

The user will be asked if they want certain ingredients in their coffee. The first question should be about espresso, if the user doesn't want an espresso drink, then the answer is filtered coffee, and the program should close after displaying the proper notification. If the user prefers espresso, then your program will ask about the other ingredients. If your program has a match for the given ingredients, it should notify the user about the created coffee kind. This means, your program should not ask for every ingredient before displaying the notification – as it should be done when just the necessary information is already entered. For the relationship between ingredients and coffee kinds, you may find a coffee flowchart below in the functions section.

You may (and you should) find the exact notification format in the "Sample Runs" below.

We will be automatically grading your homeworks using GradeChecker, so it is very important to satisfy the exact same output given in the sample runs. You can utilize GradeChecker (<http://sky.sabanciuniv.edu:8080/GradeChecker/>) to check whether your code is working in the expected way. To be able to use GradeChecker, you should upload all of your files used in the homework (**only** your\_main\_cpp file for this homework). Additionally, you should submit all of your files to SUCourse (**only** your main cpp file for this homework) **without zipping** them. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

**The name of your main source (cpp) file should be in the expected format:** "sucourseusername\_lastname\_name\_hwnumber.cpp" (all lowercase letters). Please check the submission procedures of the homework in this document.

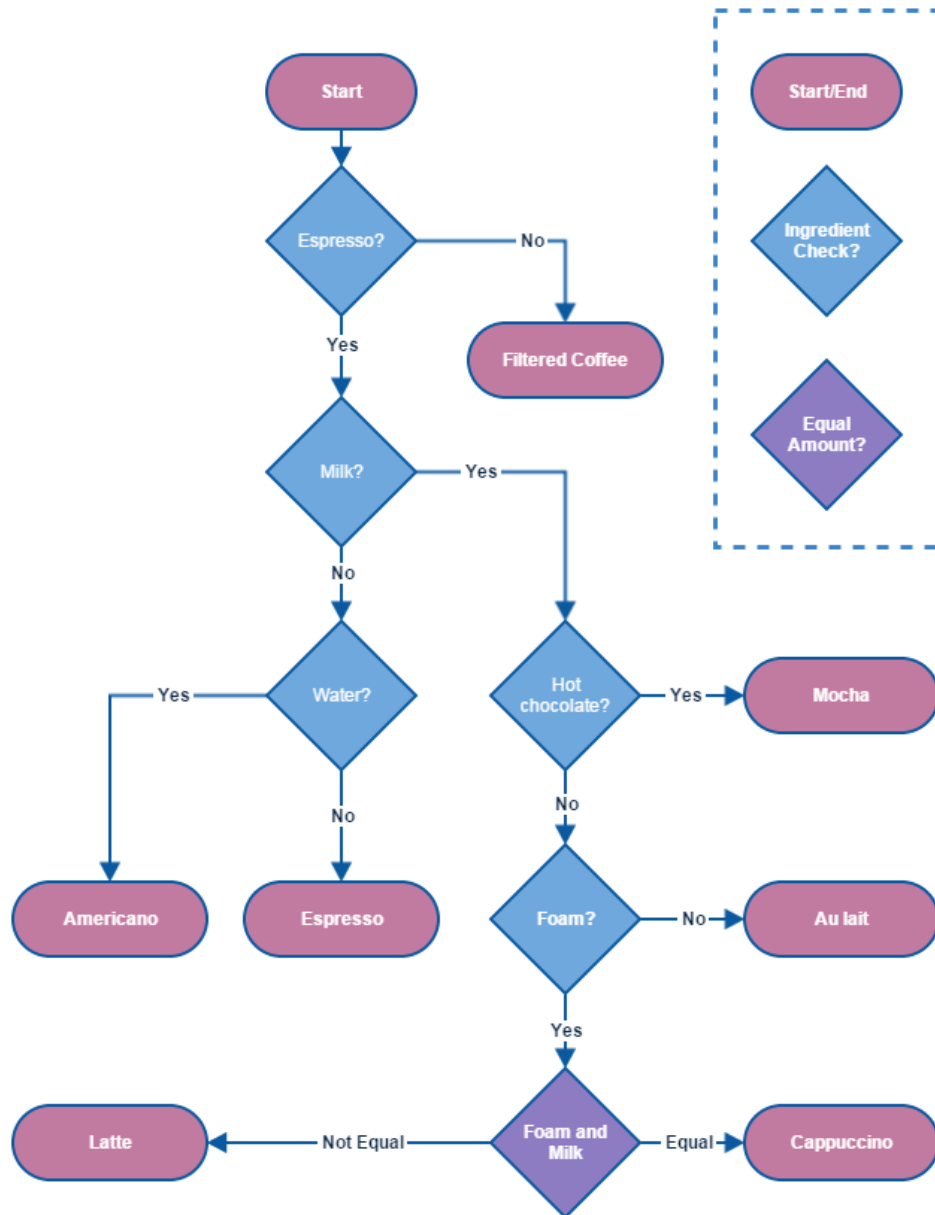
To get help using GradeChecker you may ask questions to the list of your grader TAs: [cs201gchelp@lists.sabanciuniv.edu](mailto:cs201gchelp@lists.sabanciuniv.edu)

## Input Check

You are not expected to perform any input checks. You may assume, answers to the questions will be entered only as a “yes” or a “no”. Please be careful that answers are expected in lower-case.

## Flow Chart

You may refer to Figure 1 for the flow of your program. Please note that this flow should be implemented using several functions in your program. Please refer to the functions section for details.



**Figure 1** Flow Chart

# Use of Functions (EXTREMELY IMPORTANT!)

**You have to follow the specifications below for function declaration and callings. The grading criteria will include proper use of these parametric functions. Do NOT use any global variables (variables defined outside the functions) to avoid parameter use. Unnecessary code duplication will cause grade reduction as well.**

In the first homework you were not supposed to implement any functions. However, in this homework you are expected to (actually you have to) use some function(s). The guidance about using functions in this homework is below.

A total of **four** user-defined functions (other than main) must be implemented. Two of these functions are void functions and two of them are non-void functions. You have to implement and use these four functions. If you don't, your grade will be lowered because of the missing functions. On top of these functions, you may use other functions if you want.

The program flow will be as follows:

- **main function:** You will display a welcome message, and then ask for the name of the customer. Then, you should proceed to function1 with name as its parameter. The same steps will be repeated for the other customer.
- **void function1(string):** This function will be called from main with customer name as parameter, and it will call other given functions depending on program state. Asks the questions based on the flow chart given above. First question should be about espresso, for the rest of the questions you will construct the order. For the questions, you should make use of either function2 or function3. Note that if there were more than one customer, calling this function multiple times with appropriate parameters would have been enough.
- **bool function2(string):** Prompts the user whether given ingredient is wanted or not. The string parameter will be an ingredient such as "foam" or "water". When answered "yes", this function returns true. When answered as "no", this function returns false.
- **bool function3(string, string):** Prompts the user whether the given 2 ingredients are in equal amounts or not. The string parameters will be ingredients such as "milk" or "foam". When answered "yes", this function returns true. When answered as "no", this function returns false.
- **void function4(string, string):** Takes the name of the customer and resulting coffee kind as parameters, and displays a message using them.

Needless to say, you have to name these four functions using meaningful identifiers, not as Function1, Function2, etc.

## No abrupt program termination please!

You may want to stop the execution of the program at a specific place in the program. Although there are ways of doing this in C++, it is not a good programming practice to abruptly stop the execution in the middle of the program. Therefore, your program flow should continue until the end of the main function and finish there.

## 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 order in these examples and the prompts your program will display must be **exactly the same** as in the following examples.

### *Sample Run 1*

Welcome to Pequod's Coffee! Can you tell me your name?

**Robert**

Dear Robert, would you kindly answer our questions, so we can determine the coffee you want.

Does it have espresso?

**no**

Your order is ready Robert! What you want is a filtered coffee. Enjoy!

### *Sample Run 2*

Welcome to Pequod's Coffee! Can you tell me your name?

**Alan**

Dear Alan, would you kindly answer our questions, so we can determine the coffee you want.

Does it have espresso?

**yes**

Does it have milk?

**yes**

Does it have hot chocolate?

**no**

Does it have foam?

**no**

Your order is ready Alan! What you want is a cafe au lait. Enjoy!

### *Sample Run 3*

Welcome to Pequod's Coffee! Can you tell me your name?

**Jesse**

Dear Jesse, would you kindly answer our questions, so we can determine the coffee you want.

Does it have espresso?

**yes**

Does it have milk?

**yes**

Does it have hot chocolate?

**no**

Does it have foam?

**yes**

Do foam and milk are in equal amounts?

**yes**

Your order is ready Jesse! What you want is a cappucino. Enjoy!

### **Sample Run 4**

Welcome to Pequod's Coffee! Can you tell me your name?

**Ronaldo**

Dear Ronaldo, would you kindly answer our questions, so we can determine the coffee you want.

Does it have espresso?

**yes**

Does it have milk?

**no**

Does it have water?

**yes**

Your order is ready Ronaldo! What you want is an americano. Enjoy!

## **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 clarifying 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:

- ☐ **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/](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 .cpp 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.

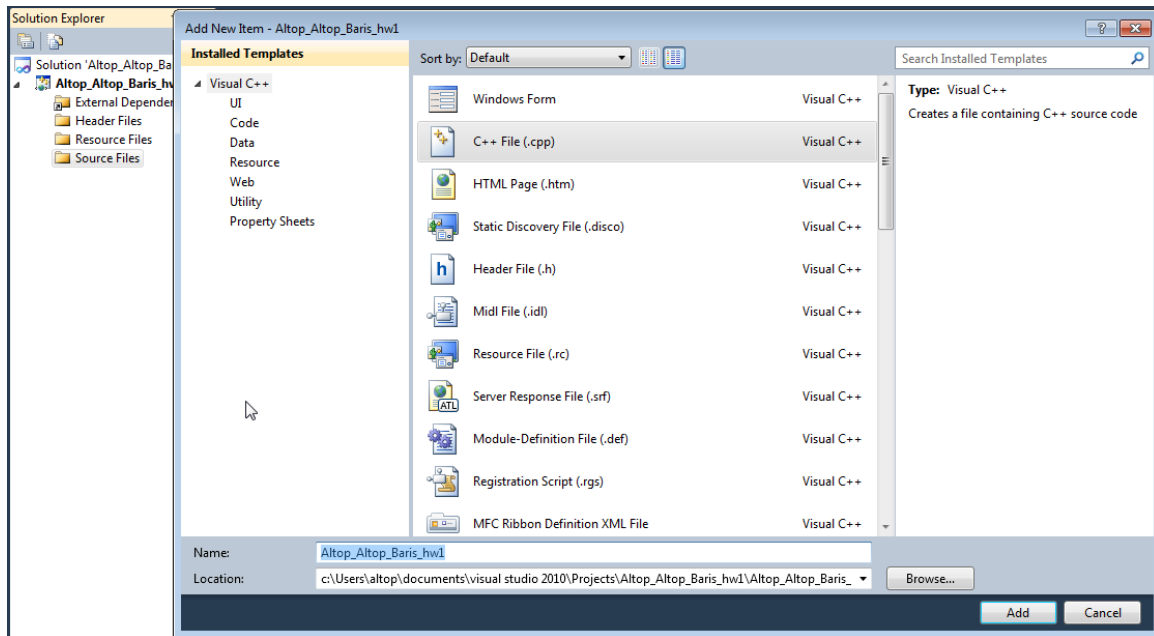
## What and where to submit (IMPORTANT)

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 Özbugiszkodyazaroglu, then the file name must be:

**cago\_ozbugiszkodyazaroglu\_caglayan\_hw2.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.
- ❑ Check that your **cpp** file opens up correctly and it contains your source code. You will receive no credits if your cpp file cannot be opened, empty or it does not contain the correct solution.

### **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 2 in the assignments list.
  - 3) Click on "Add Attachments" button.
  - 4) Click on "Browse" button and select the cpp file that you generated.
  - 5) Now, you have to see your cpp 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 2 in the assignments list.
- 3) Click on "Re-submit" button.
- 4) Click on "Add/remove Attachments" button
- 5) Remove the existing cpp file by clicking on "remove" link. This step is very important. If you do not delete the old cpp file, we receive both files and the old one may be graded.
- 6) Click on "Browse" button and select the new cpp file that you want to resubmit.
- 7) Now, you have to see your new cpp 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!***

***Hanefi Mercan and Gülşen Demiröz***