

## CS201 – Spring 2015-2016 - Sabancı University

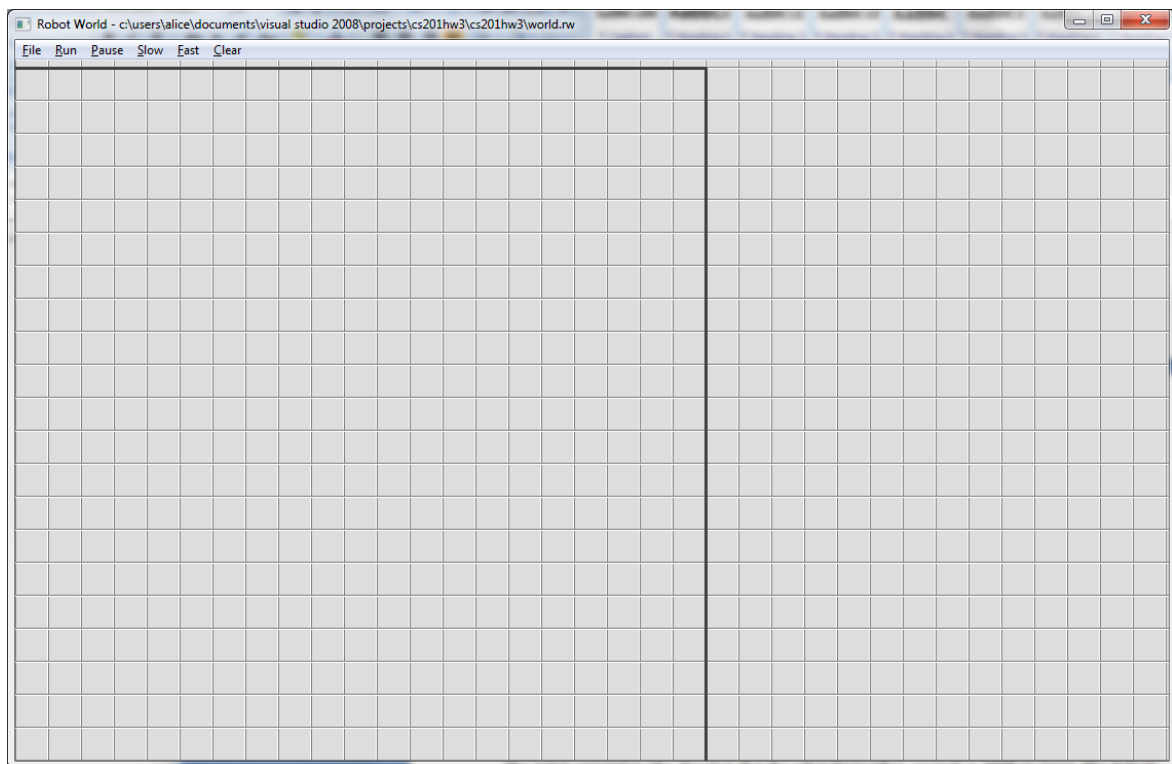
### Homework #3: Who's a good boy?

**Due November 2, Wednesday, 19:00 (Sharp Deadline)**

#### Brief Description

In this homework, you will write a robot application that uses the Robot class discussed in the lecture. This class is also described in a document called “RobotWorld.pdf” that is in the robot.zip package available in CS201 website. All the material needed for the robot class is also in the robot.zip (at the website of CS201). Please examine the RobotWorld.pdf file and the example programs. You will practice on the robot class during recitations this week. To do this homework, it is to your benefit to attend recitations because Robot World is something quite different considering what you have learned until now.

The C++ program that you are going to write will create two robots (owner and dog) in a location of which the x and y coordinates are inputs. The world is 21x21 square so keep in mind that x and y values cannot be bigger than 19. The homework has two goals. First the dog must move to left cell of the owner. Second goal is to move the owner to (10, 20) and the dog to (9, 20). Details will be explained later in the document. A screenshot of the empty world is given below:



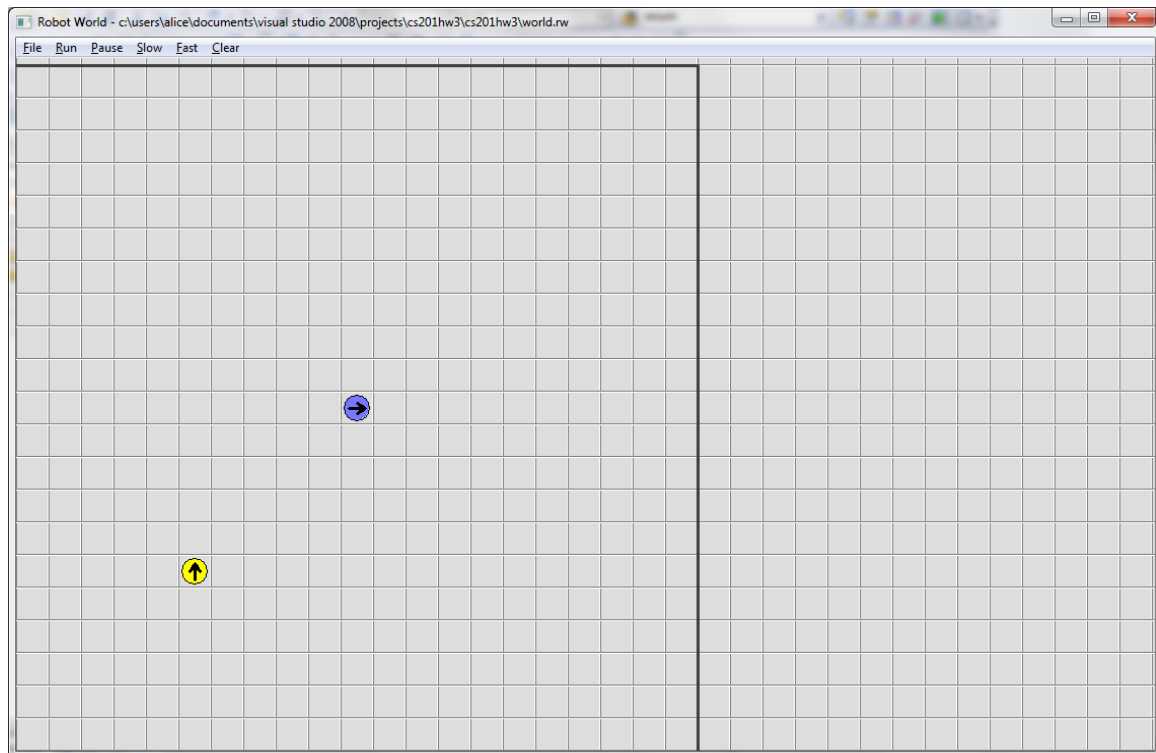
There are some limitations for x and y coordinate inputs for owner and the dog. They will be explained in a later section.

## Details and Rules

First, the position of the owner will be input by the user. The coordinates of this robot must be a non-negative integer larger than (but not equal to) 0 and smaller than (but not equal to) 20. Otherwise your program should display an appropriate error message and should not even create the robot (and of course no robot movements will be performed). The initial orientation (the direction that the robot faces when created) is NOT an input. The default orientation is east and the robot is to be created facing North. The default color of the robots after creation is yellow.

Second, the position of the dog will be input by the user. The coordinates of this robot must be a non-negative integer larger than (but not equal to) 0 and smaller than (but not equal to) 20. **In addition, the x-coordinate of the dog must be larger than x-coordinate of the owner.** Otherwise, your program should display an appropriate error message and should not even create the robot (and of course no robot movements will be performed). The initial orientation (the direction that the robot faces when created) is NOT an input. The default orientation is east and the robot is to be created facing east. The default color of the robots after creation is yellow. **You should change color of the dog robot to blue.** You can use SetColor() method of given Robot class to change your robot's color.

Your world may look something like this when the robots are first created:

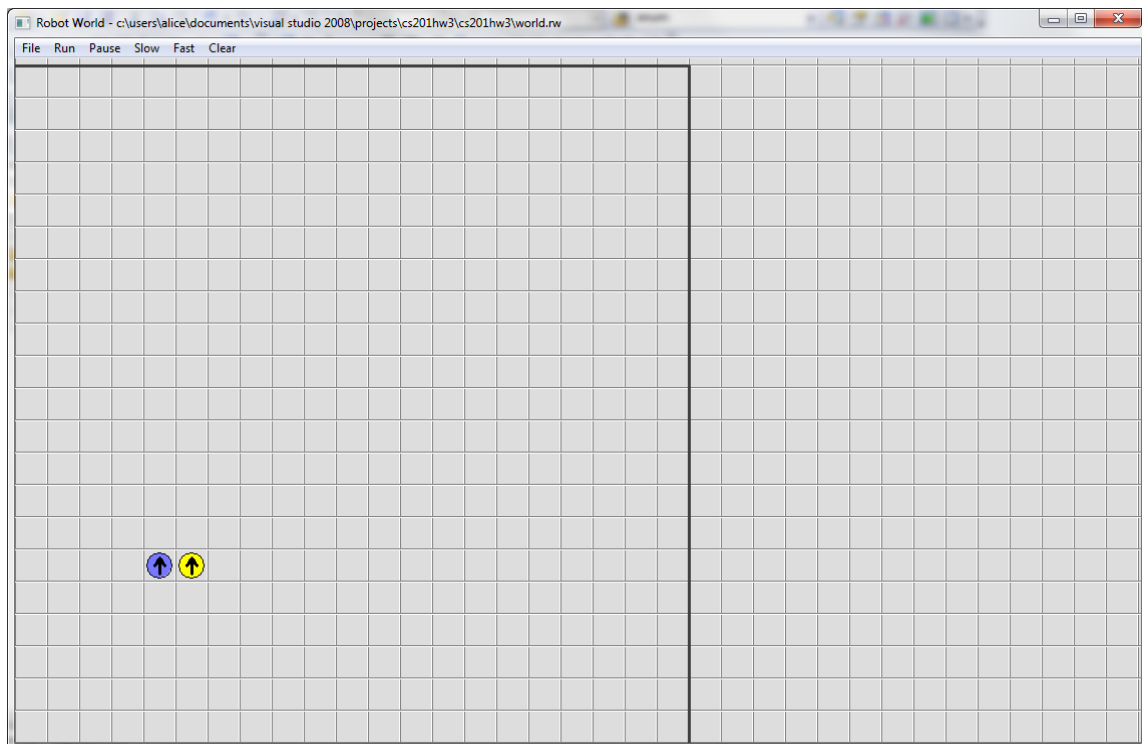


First goal is to move the dog to the cell left of the owner. **Remember, the dog will move first.** You should calculate number of cells the dog must move in x and y directions. In the end, the dog should be positioned to the left of the owner, and must face north.

### **IMPORTANT!**

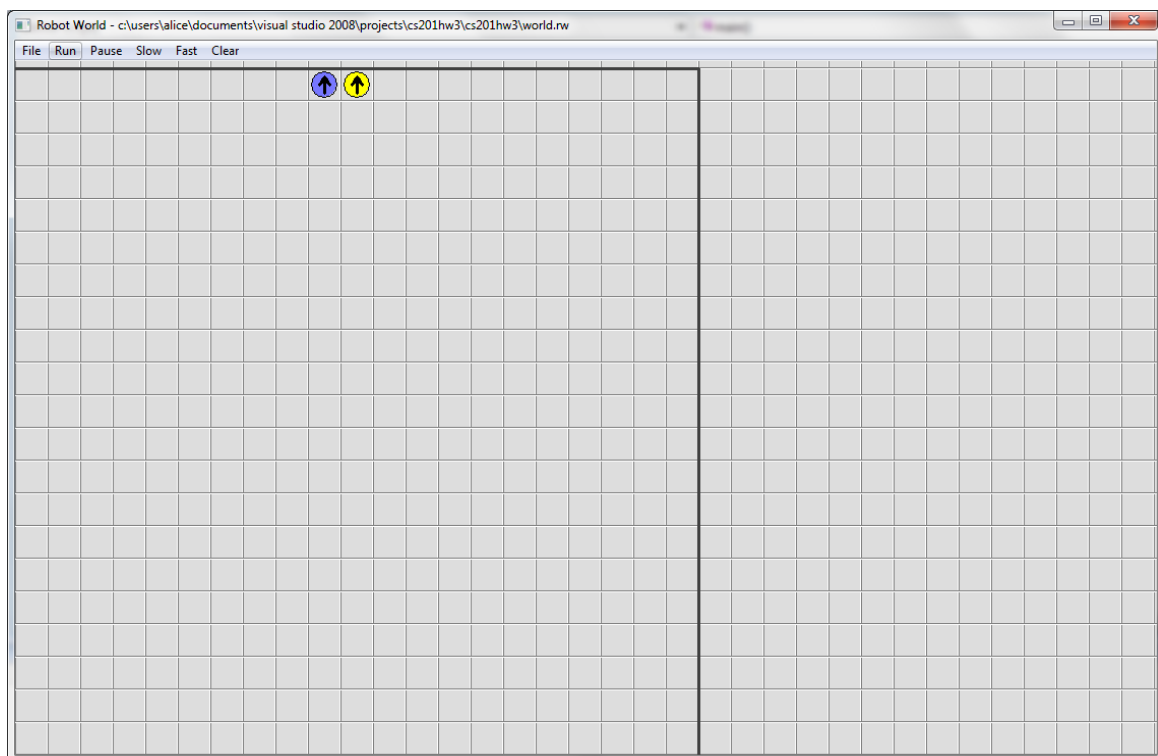
If your code does not compile, you will get **zero**. Please be careful about this and double check your code before submission.

See a screenshot below (owner yellow, dog blue):



After the dog is moved to the left of its owner; second phase of the movement begins. The phase will end when the owner moves to (10,20) - facing north; and the dog moves to (9,20) – facing north.

See a screenshot of the end result:



Hint: In order to avoid any collisions, you may move the owner first along the y-axis and then the dog along the y-axis. After both robots are positioned, you should move them along x-axis.

## Input Entry and Input Check

- Your program should get and check x-coordinate of the owner. This input should be between  $0 < x < 20$ .
- Then your program should get and check y-coordinate of the owner. This input should be between  $0 < x < 20$ .
- Afterwards, your program should get and check x-coordinate of the dog. The x-coordinate of the dog should be larger than the x-coordinate of the owner and  $< 20$ . Finally, your program should get and check y-coordinate of the dog where  $0 < \text{y-coordinate-of-the-dog} < 20$ .

If any of the inputs are not valid, your program should end without any robot creation or movements.

## Input and Output in Robot Environment

You cannot use `cin` and `cout` for input and output in robot environment. The reasons for this fact and remedies are explained in the RobotWorld.pdf file. Please read this document carefully and examine the example program given in recitation materials this week in order to understand the I/O functionality that you are going to use in this homework.

## Use of Functions and Other Rules

Unlike the second homework, we will not specify any functions here. But you are expected to use functions to avoid code duplication and to improve the modularity of your program. **If your main function or any user-defined function is too long and if you do everything in main or in another user-defined function, your grade may be lowered.**

**AND PLEASE DO NOT WRITE EVERYTHING IN MAIN AND THEN TRY TO SPLIT THE TASK INTO SOME FUNCTIONS JUST TO HAVE SOME FUNCTIONS OTHER THAN MAIN. THIS IS TOTALLY AGAINST THE IDEA OF FUNCTIONAL DESIGN AND NOTHING BUT A TRICK TO GET SOME POINTS. INSTEAD PLEASE DESIGN YOUR PROGRAM BY CONSIDERING NECESSARY FUNCTIONS AT THE BEGINNING.**

Try to use parametric and non-void functions wherever appropriate. Do NOT use any global variables (variables defined outside the functions) to avoid parameter use.

## Demo Application

Due to interactive feature of this homework, we are not able to provide sample outputs, but we provide an executable file that can be run to understand how your program should work. We have already written the program for this homework and the corresponding executable file (hw3.exe) as well as the world file (world.rw) is given to you within the same zip package of this homework document.

## 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.

## 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.

### 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 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/](http://myweb.sabanciuniv.edu/gulsend/su_current_courses/cs-201-spring-2008/plagiarism/)

<b>Plagiarism will not be tolerated!</b>
--

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.

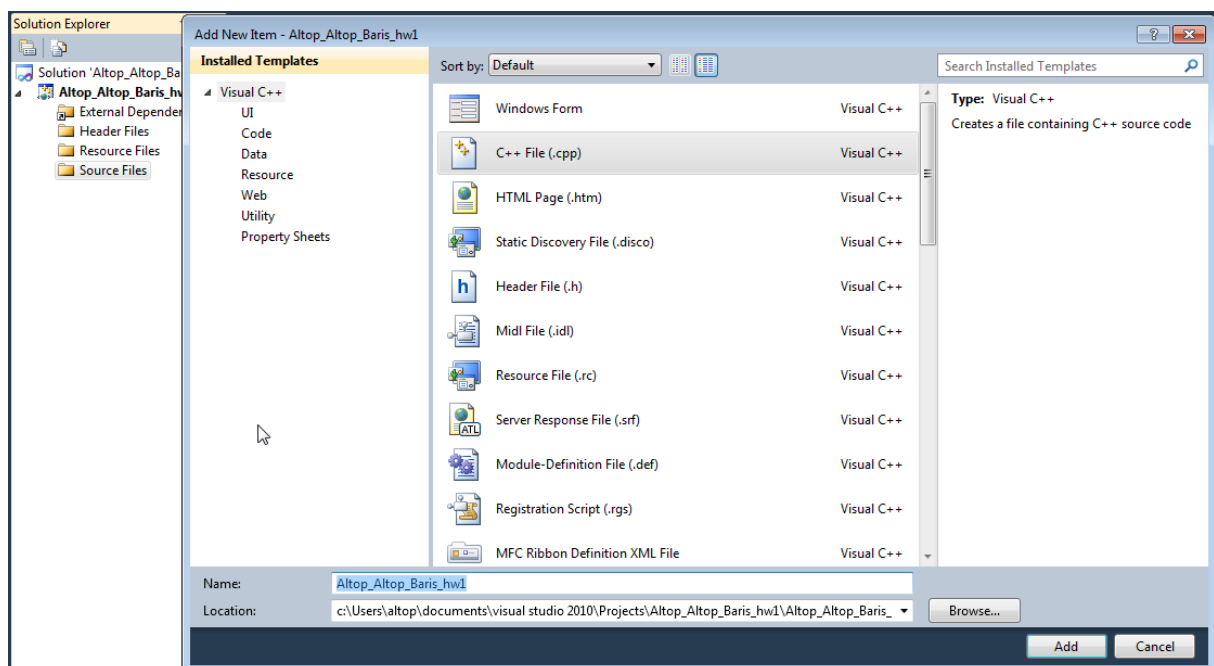
## What and where to submit (IMPORTANT)

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:

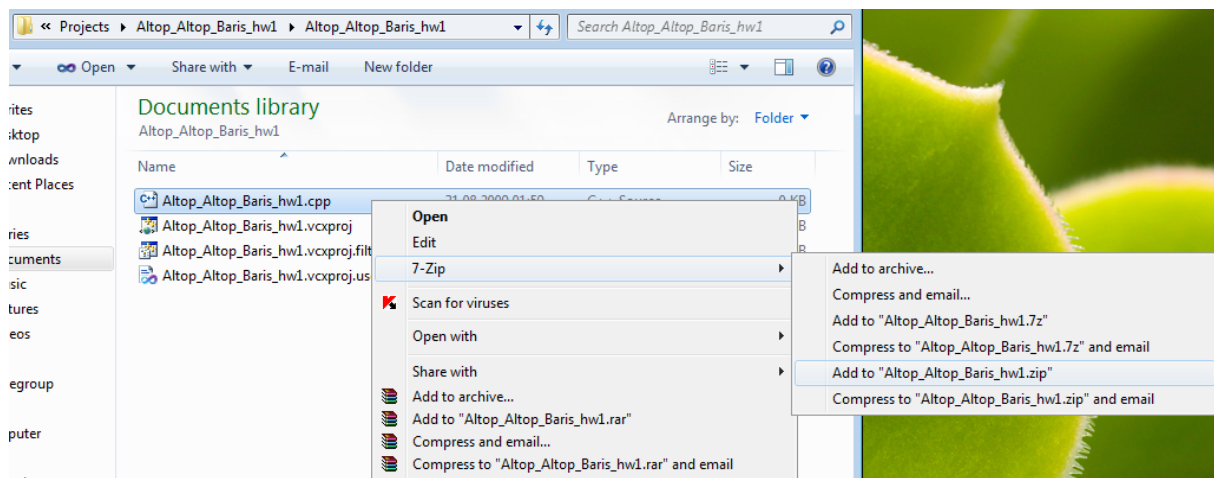
- ☐ Use only English alphabet letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
- ☐ Name your cpp file that contains your program as follows.  
“SUCourseUserName\_YourLastname\_YourName\_HWnumber.cpp”



- ☐ 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\_hw3.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.
- ☐ You need to submit ALL .cpp and .h files including the robot and minifw files in addition to your main.cpp in your VS solution. Compress all your necessary files 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\_hw3.zip is a valid name, but hw3\_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 3 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 3 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*