# Computer Programming Homework Trivia

## *Instructor: Dr. Aleksander Malinowski*

### Problems and their solution

Each homework assignment will be posted on the course web site even if they are not announced in class each time. Typically one to two programs to write, modify or correct are requested. You may try to use ideas illustrated by examples discussed in class and posted on the course Web site and merge several of them together to form a program that works towards the solution. You would still need to do many modifications to meet requirements of the requested functionality.

In case of this course, for the sake of your learning experience use of code from your classmates or older friends who took the course in past year is at least discouraged. You need to learn how to come up with your own solutions and then how to code them. Otherwise, you would act merely as a technician. While graders are instructed not to ask as copyright policemen who fight against plagiarism, the exams are designed to measure your understanding of subject and proficiency in problem solving followed by proficiency in coding your algorithm. The last two skills can be learned mostly by exercising problem solving and rarely only by studying existing solutions. **While simply doing all homework cannot guarantee top grade in the class, not doing it, or doing with excessive outside help is a recipe for not learning the material and failing the course**.

### Making your program working

Your first task is to make the program work towards requested solution. Programs unrelated or only remotely related to the assignment receive no credit even if they work perfectly. After writing the program, you need to compile it using one of available C++ compilers. In order to be able to run a program you have to be able to compile it without any errors and (preferably) without warnings. That is required to get any significant credit for your work. At this moment, still less than half of the work is done. You need to run your program up to several times using different data as user input and check if it works correctly. If it does not - study your program, analyze what it really does and modify it. Remember that **the computer will always do exactly you tell it to do in your program, not what you want it to do.**

In a way writing software can be compared to writing an essay. There are so many words you can use! However, only certain combinations make sense. **It is unlikely to make a logical sentence from random words, so please do not try to correct your program by means of random experiments. Try to analyze what it does and compare that with what you want it to do.** Random changes will most likely add to the mistakes rather than correct them.

When your program finally works correctly you have to document that by saving the program output. Depending on the course and homework that might by simply computer screen dump, contents of a file, and/or highlighting of added code in provided partially working program. If you forget to do so a fee of 10% to 20% of points will be charged.

### Removing compiler errors

Sometimes one small error or typo can generate many error messages. For example, a mistake in the variable declaration may cause an error message in each line in which that variable is used, even correctly. On the other hand, serious errors may generate lesser number of messages simply because the compiler would give up completing its work. **Do not panic or get discouraged by numerous errors pointing to many lines of your code and sometimes described in a cryptic way. Usually inspection of the first line with error and a line above would reveal one problem reported several times, and after you get some experience, they will not be cryptic anymore.** Try to analyze first error and inspect the line the message points to. Sometimes the error can be found in one of a few proceeding lines, for example missing semicolon. Before you acquire more experience, you may want to remove only one error at a time and repeat the compilation. In that process sometimes the number of reported errors would increase, as eliminating serious mistakes would allow the compiler to look for smaller problem that were initially overlooked due to incorrect interpretation of your code after the first mistake.

### Dealing with warning messages

Warning message is the way compiler lets you know that you may have done something incorrect even though the instruction is technically correct. You may consider a warning as a possibly logically incorrect sentence without any grammar or spelling error. The issue of common warnings will be discussed in the lectures. The best thing to do is to avoid warnings.

### Submitting the assignment

Homework assignments have to be submitted using the Sakai course management system. There is a separate drop box for each assignment. The due date is typically 11:55pm on Friday evening. Late submission is accepted with "late fee" deducted from the grade as described in course grading policy.

### Sample solutions

There are many examples of features exercised in homework illustrated by short programs discussed in class. Since every student's work can be unique no single sample solution would address all questions. Solutions are posted rarely.