

The Guidelines for submitting a programming assignment problem are outlined in this document. For the remainder we assume you are submitting an implementation outlined in Programming Assignment hypothetically named A and the last four digits of your ID (NOT YOUR SOCIAL SECURITY NUMBER) are hypothetically WXYZ.

## 1 Why C++ or C and not Java?

For your implementations you can use C++ or plain C. You can not use Java (even if you used Java in your CIS 113/114 which is highly unlikely if you take CIS 435 in year Spring 2006 . In the remote possibility that this happened we offer our justification: CS students should be proficient in both C++/C and Java. If you used Java before, it's time to practice more C++/C; if you did not learn Java before, so much the better: we are not going to ask you to use it. As far as the C++ vs C question is concerned, the C++ language subsumes C. We offer a non-traditional interface for our programs to reduce the temptation of you looking up for code on the Internet.

## 2 Email Format

For each programming assignment we expect you to submit an email message with subject line `paAsWXYZ`, with `pa` identifying the email as programming assignment A from sender WXYZ. The subject line should look like. The email should be sent to `alg435@cs.njit.edu`. If you send it to any other email address it will be ignored.

Subject: `paAsWXYZ`

A SINGLE ASCII text file or an attached ASCII text file named `paAsWXYZ.c` for a C program or `paAsWXYZ.cc` for a C++ program by email to `alg435@cs.njit.edu` that conforms to the following requirements.

- The first line of the file must contain your name and the PA number.
- The function that will be implemented and possibly other functions that you use should follow.
- The name of the function to be implemented must be the one indicated in the description of the programming assignment.
- The file that you send us MUST NOT CONTAIN a `main()` function. Do your testing separately; we are not interested in it.
- Experimental results related to the assignment should be in the form of C not C++ comments.
- You must make sure that the file you submit compiles as a standalone C++/C function. If it does not compile, we will not attempt to edit it to make it compilable. Your submission will be rejected.

```
/* Alex. Gerbessiotis PA1. */  
// <<<< Do not use C++ oriented single-line comment lines.  
/* Use multi-line C oriented comment lines like this  
   one */  
int int_square (int x)  
{  
    return(x*x);  
}  
  
/* Experimental results are included here. This line  
   * may extend on multiple
```

```
* lines such as this
* example
*/
```

In the handout section, function `bubble.c` offers a bubble-sort (unoptimized) implementation that conforms to the guidelines you will observe for implementing sorting functions. This standalone function can compile into a standalone executable file in C or C++.

### 3 Implementation guidelines

Make sure that the entry points of your functions adhere to the format outlined in the programming problem description.

1. *It is your responsibility that your code is ANSI C++ or ANSI C compatible and compilable. We can only tell you that testing will be done on a SUN Workstation or a Linux-based PC or a Windows based PC using gcc/g++ in the first two cases and gcc/g++ or Visual C++ in the latter case. Which platform we will use for what assignment, we will not reveal it to you.*
2. *No partial credit will be given to submitted code that does not satisfy the previous guidelines. No partial credit will be given to submitted code that does not compile. If your code fails one of the guidelines, we will not request retransmission.*
3. *No partial credit will be given for code that does not fully list its bugs. The grader will decide testing instance(s) and grade your submission based on whether it passes successfully or not these testing instances. If your code does not pass any testing instances, it will get 0 points.*
4. *C versus C++* Some of you may complain that `malloc/free` are C and not C++. They are part of any C++ compiler. If you don't know how to use them, then you will learn a little more about C++ that you didn't know. If you already know how to use them, then you can practice more with their usage. On an AFS account `man malloc` may provide some extra documentation. We intentionally use `malloc/free` as opposed to `new/delete` to make the assignments more interesting, and also incompatible with published web-available code.
5. **Assistance. Help. Manual pages.** If you don't know the syntax of a C++/C command, on an AFS system you can look it up using the `man` command. Or you can use the Visual C++ help system.
6. *Visual C++ vs other environments* You are free to use whatever environment you like. If your code is ANSI-compliant it should compile everywhere with any ANSI-compliant compiler. If you follow our guidelines then you can use the Visual C++ environment to write your ANSI C++ or ANSI C compatible code and test it for example on AFS under `g++` or `gcc` to check for compatibility problems.
7. *How to use Visual C++ and Microsoft's IDE.* We do not provide assistance. You should know by now. This is CIS 113 material not even CIS 114 but definitely not CIS 435.
8. *Debugging.* Every junior or senior year should know how to debug a program.

### 4 Supplied Code

In the course Web-page you may find a file named `testing.tar`. It is a tar archive. You can extract the files using `Winzip` under MS Windows.

In a Unix (eg. AFS) system, if you type in the command prompt %

```
% tar xvf testing.tar
```

you can extract the files of the archive.

The three files of this .tar file, `Makefile`, `bubble.c` and `sortg.c` are also available individually.

The source code of the second file `bubble.c` also appears in one of the homeworks. `Makefile` shows how to compile and link these files under Unix/Linux. An example of a sorting algorithm (bubble sort) is provided in `bubble.c`. This example also shows how the arguments that are common to all functions of this assignment operate. File `sortg.c` shows how to call the function in `bubble.c` and how to define a specific `compare` function (See discussion in part A of sorting oriented programming assignments when they become available). Although in part B of the programming module you are going to test your algorithms on input arrays of integers, the testing functions of the grader may test your code on sequences whose elements are `double`, `float`, strings, etc.

*It is imperative that you not make any assumption on the data type of the input sequence.*

**Makefile.** If you don't know how to use a Makefile you can import `sortg.c` into your Visual C++ project and let do as you please with it without the Makefile. If you use AFS it will be of assistance to use the Makefile. It is NOT a requirement that you use a Makefile! If you don't know anything about make, type in AFS `man make` and you will find out more. Note that having `makefile` and `Makefile` files at the same directory is dangerous and/or confusing to you and the `make` program. The AFS `make` is GNU `Make` that is freely available under the GNU public license. You can get extensive information about it through the Web as well. Visual C++ has a `make` program called `nmake.exe`. Visual C++ projects use it for compilation. You can compile code in Visual C++ through a `makefile` as well; don't ask us how, however. Read the help topics in Visual C++ or look up for additional information on the Web.

## 5 Programming Assistance

We respond only to questions strictly related to the clarification of the requirements of the programming assignments. If you don't know how to use Visual C++, this course will not teach you how to use it or how to program in C++. CIS 113 and CIS 114 and the new CIS 288 might.

You are supposed to know how to use some environment to program in C++. We don't care whether you program on a PC with Visual C++, on a Linux PC using `gcc`, or an AFS machine using whatever compiler it is available there. The end result is that your ANSI-compliant code should be compilable using our ANSI-compliant compiler.

**Segmentation Faults/Core dumps** If you get a related message, then your program has a bug that is pointer related. Debug it! Don't ask us how. You are supposed to know by now. You are a junior or senior CS student who has already taken CIS 113, CIS 114, CIS 280 and other programming oriented courses. Every CS major should know how to debug a program by now! ■