The Guidelines for submitting a programming-based homework problem are outlined in this document.

For the remainder we assume you are submitting an implementation outlined in a fictitious Homework numbered `A` and the last four digits of your ID (NOT YOUR SOCIAL SECURITY NUMBER) are `WXYZ`. If you do not know your ID make sure that you find it out. It will be required in all exams.

1 C++ or C or Java?

For your implementations you can use C++ or plain C or Java. There will be some assignments that can only be done in C++ or C. Therefore plan ahead carefully. We can guarantee that two of the assignments will be possible to be done in Java and all of them in C or C++. You are not allowed to use classes/packages that are external to the Java language. All the source code you provide us must be yours. No libraries/packages (in the form of `jar` or `class` files) will be accepted. Similar guidelines apply to C++ or C users. Make sure that your supplied code COMPILES and RUNS at an afs machine. One of `afs10` through `afs20` will probably be used for testing of Java, C++ or C code.

If your submitted code does not compile on the test afs machine, you will get 0 points. We will not ask you to resubmit. If your program crashes and your supplied documentation DOES NOT include a list of bugs and abnormal program behavior you will get 0 points as well.

Again note, that you only need to collect 167 points from the programming assignments. This is roughly the work of two programming assignments.

2 Email Format

For each programming assignment we expect you to submit an email message with subject line `hwA-WXYZ`, with `hw` identifying the email as homework `A` from sender `WXYZ`. The subject line should look like.

```
Subject:  hwA-WXYZ
```

The email should be sent to `alg610@cs.njit.edu`. If you send it to any other email address it will be ignored. You need to write the email address alg610@cs.njit.edu from scratch. Do not reply to a previous email of ours. Handout 0 offers an explanation on why this is so. Note also that alg610@njit.edu or alg610@oak.njit.edu ARE NOT VALID ADDRESSES. The NJIT mail subsystem only recognizes alg610@cs.njit.edu.

A SINGLE ASCII text file or an attached ASCII text file named `hwA-WXYZ.c` for a C program or `hwA-WXYZ.cc` for a C++ program or `hwA-WXYZ.java` for a Java program by email to `alg610@cs.njit.edu` that conforms to the following requirements is required of you. You may provide a second attachment that lists bugs or provides not trivial instructions and information in a attached file of the form `hwA-WXYZ.txt`.

NOTE. If the file-names of the attachments deviate from these guidelines your submission will be rejected. For the first time test by sending an email to you and see whether everything worked fine.

If you have more than the two files above, archive all of them into a `hwA-WXYZ.tar` tar-file or a `hwA-WXYZ.zip` zip-file and send that file as an attachment.

- The first line of each file must contain your name and the HW,ID numbers.
- 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. If they are not WE WILL NOT FIX THEM. IT'S YOUR RESPONSIBILITY. YOUR WILL THEN BE IGNORED AND RECEIVE 0 POINTS.

- Experimental results related to the assignment should be in the form of C or C++ or Java comments.

An example is given below.

```
 /* Alex. Gerbessiotis  PA0.  */
 /* Use multi-line C oriented comment lines like this
    one */
 // You may also use C++ -style
 // line comments
int int_square (int x)
{
 return(x*x);
}

 /* Experimental results are included here. This line
  * may extend on multiple
  * lines such as this
  * example
  */
```

3 Implementation guidelines

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++.

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

1. You must make sure that the file you submit compiles as a standalone C++/C function or a JAVA class. If it does not compile, we will not attempt to edit it to make it compilable. Your submission will be rejected. Certain compilers (or careless programmers) attach to a source file header-file include directives that are specific to a given compiler. If you compile your code with that compiler, everything works ok; if however we use a different compiler your code will not be compiled properly. Thus we insist on you writing ANSI C++/C compliant code. One easy way to test whether this can become a problem is try to compile your code on a different (say, AFS) machine with a different (say, non Visual C++) compiler. **MAKE SURE YOU CODE COMPILES ON AFS MACHINES AFS10 THROUGH AFS20.**

2. *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.* For JAVA we will use the same platforms and probably an AFS machine with the currently installed JAVA compiler. Which platform we will use for what assignment, we will not reveal it to you.

3. *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.*

4. *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.*

5. *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.

6. **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.

7. *Visual C++ vs other environments* You are free to use whatever environment you like. If your code is ANSI-compliant it will 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.

8. Java Compatibility. Do not use classes that are not supported by standard Java. Sun's implementation changes from one version to the other. Test your code on at least two AFS machines to determine full compatibility.

9. *Debugging.* Your are a CS student, you should know by now how to debug your program with your favorite tools in Visual C++ or whatever system you are using. If everything else fails, you can still use cout or printf statements in your code to identify buggy code (and printf statements also work in Java now).

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 (AND WILL) 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 C, C++, Visual C++, or Java, this course will not teach you how to program in C/C++/Java. CS 505 is such a course; you either learned so in that course, or you convinced admissions that you had this background and they waived that course requirement for you.