1 Introduction

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 a fictitious Programming Assignment 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 class exams.

The NJIT grading system records only student ID's. Therefore if you provide us with your SSN digits you will confuse us. Your SSN is a private piece of information. We WILL NEVER ASK YOU TO REVEAL IT TO US.

2 C++ or C or/and Java?

For your implementation 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++. For java, you are not allowed to use classes/packages that are external to the Java language. All the Java code you provide is source code in the form of `.java` files and it is yours. No libraries/packages (say in the form of `jar` or `class` files or some other format) will be accepted.

Similar guidelines apply to C++ or C users.

For either C++/C or Java code, make sure that your supplied code COMPILES and RUNS (if it is working) on an afs machine. One of `afs10` through `afs20` will probably be used for testing of Java, C++ or C code.

3 Program Submission guidelines

3.1 Email Format

We use the convention outlined in the Introduction of this document. For each programming assignment we expect you to submit an email message with subject line `paA_WXYZ`, with `pa` identifying the email as a programming assignment with number the fictional `A` from sender the fictional student whose last four digits of his/her id are `WXYZ`. Whether you use `_` underscores or `-` dashes, the choice is yours. For C/C++ programmers there is no difference; Java programmers might wish to avoid the use of a dash altogether.

The subject line should look like.

```
Subject:  paA_WXYZ
```

The email should be sent to `alg610@cs.njit.edu`. If you send it to any other email address it will NOT reach us. You need to write the email address alg610@cs.njit.edu from scratch. Do not reply to a previous email of yours (or 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`.

3.2 File names and Java function names

For most cases of C or C++ code submission, A SINGLE ASCII text file or an attached ASCII text file named `paA_WXYZ.c` for a C program, or `paA_WXYZ.cc` for a C++ program would suffice. For a JAVA program, you can use `paA_WXYZ.java`. If you decide that further explanation to the grader is needed you may also attach a `Readme_pA_WXYZ.txt` ASCII text file with instructions or explanatory information or a potential bug report.

Note that in files you are asked to use underscores and not dashes. Underscores _ are compatible with both Java and C/C++ naming conventions. If a function name in a homework has an underscore in it and you plan to implement it in Java, you might replace it with an underscore.

3.3 Information in files

In the `paA_WXYZ.c` or `paA_WXYZ.cc` or `paA_WXYZ.java` files and any other file you submit record the following information at the very very beginning.

- The first line of the file must contain your name and the PA number in the form of a comment.
- The function that will be implemented and possibly other functions that you might 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
  */
```

4 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.

4.1 Compilation

We insist on you writing ANSI C++ or ANSI C compliant code. We insist on you writing Java version independent code. You must make sure that the file you submit compiles as a standalone C++/C function or a JAVA class. Do not use additional functions/classes that might be language version dependent.

To make sure that your code is not machine dependent, try to compile it on a different machine. For example you can test compile your code on **one or more AFS MACHINES.** If your code 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 directives (in the form of header files) 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 (or version of) your code will not be compiled properly.

`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.

4.2 Testing platform

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.

5 Grading

`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.
`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.

5.1 Assistance/Questions

We respond only to questions strictly related to the clarification of the requirements/semantics of the programming assignments. We do not provide programming assistance. You are a graduate student, you should know programming by now. that purpose.

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.

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.

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.

Debugging. You are a CS student and thus 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).

5.2 Supplied Code for C/C++ implementations

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 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.