# Assignment 1 for CPTN230

**Name:** Analyze and Professionally Document a Working C++ Application and Class

**References:**

* Text book
* Notes from CPTN175
* Internet Help Sites
* Anything else except other people, this an individual assignment

### Assignment Overview:

As a software developer we are often required to “figure out” and “present” the details of an application we have never seen before. The information must be formally documented using an organization’s documentation procedures. This assignment will require you to accomplish this type of task using NCC resources and CPTN230 methods. The ability to correctly and completely analyze code develops good troubleshooting skills for a failing application. This skill is essential to the timely development and maintenance of software applications.

The requirements for this assignment are exact and no deviation from the requirements or substitution of requirements is allowed. They would not be allowed in a software development organization so we will not allow them here.

The open hidden agenda items for this assignment are:

* Demonstrate the ability to analyze a C++ application
* Demonstrate the ability to use a C++ software development environment
* Demonstrate the ability to use Blackboard
* Demonstrate the ability to read a .pdf file
* Demonstrate the ability to use Microsoft Office Word 2007
* Demonstrate the ability to professionally document a C++ application
* Demonstrate the ability to manage your time
* Demonstrate the ability to follow an organizations procedures and policies

The basic steps needed to complete this assignment are listed below. The “Assignment Description” takes each of these steps and expands on them. You should consider the “Assignment Description” section as an extension of the assignment requirements. The “Strenuously Recommended Approach” section lists steps, suggestions and ideas on how to complete the assignment. In fact, each section builds and expands the assignment requirements.

1. Download all files from Blackboard.
2. Compile, Link and Run the Application.
3. Document the Procedure for Creating and Running the Application.
4. Analyze and Document the C++ Source Code and Header files.
5. Post the Results to the Blackboard via “View/Complete Assignment” link.

As you read this document you might think (or say), “Is he crazy? How in the world will I get this incredible amount of detailed work done on time?” It is very doable and you should already know how to do all the steps. If not, this is your chance to figure it out with help from me. The techniques used for this assignment will be consistently used throughout the rest of the assignments and the project. The application I ask you to analyze is not that large. This document is probably larger than the one you will create for this assignment.

### Assignment Description:

1. Download all files from Blackboard.

Login to blackboard and down load all the files associated with this assignment. There are 7 files and they are:

* This document
* 3 .pdf files. 1 for each of the software modules used in this application
* 2 .cpp files. 1 for the application source code and 1 for the class source code
* 1 .h file which is the class header file

Store these files on the system or systems you plan to use for this and future assignments. This will allow you to view them any time without having to go back to Blackboard. You can delete them later, but I wouldn’t. Space is provided on the NCC system (share drive) but you may also use your own personal systems.

1. Compile, Link and Run the Application.

Create and run the application from the provided .cpp and .h files. NCC provides the needed software development environment for creating applications from multiple C++ source and header files. You should be able to do this from CPTN175, but I am willing to assist for this assignment. As you do this, take detailed notes or start your documentation. See “Assignment Deliverables” for the document content requirements. Do not rely on memorizing or remembering the steps and then later documenting the procedure.

As before, you may do this on your own personal systems but I will not help you resolve problems encountered on your personal systems. I will only help resolve problems on the NCC systems. Working with your personal systems introduces too many potential issues to successfully troubleshoot a problem.

1. Document the Procedure for Creating and Running the Application.

From your detailed notes, unless you already started the document, write down an extremely detailed step by step procedure to create and run the application.

First, list what operating system you are using (version and service pack level) as well as the name and version of which software development environment you are using.

The actual procedure should tell me how to start the development environment, what items to select, click or check in the correct order and finally how to save files and close the environment. It should include exactly, what window, sub window, panel, pane or frame I should be using at any given step. Use the correct terms and don’t miss any steps no matter how trivial. Add notes of explanations or observations as you see fit. This is why you should not try to do this from memory. You will forget things.

You should assume your audience (me) has no clue as to what I am doing but I could still follow your procedure. I have access to many different software development environments on many different types of systems and am quite likely to have my wife try to use your procedure. If she cannot follow the procedure, it is not detailed enough.

When this assignment is done you might want to take just this section of the document and create a separate document for software development procedures. As this course proceeds and beyond, it can be a very useful tool for creating applications later. I’m not kidding. To develop the code for this assignment I used my personal laptop. At one time I had not used the software development environment on my laptop for over a year and when I started it up, I could not remember how to create an application with this environment. Yes, I would have eventually figured it out after a few hours of playing, looking up help and/or asking someone but since I wrote a “how to” document I was writing code within 15 minutes. The time spent writing and testing the code for this assignment was a different story.

1. Analyze and Document the C++ Source Code and Header files.

Details covering this step are covered in the “Strenuously Recommended Approach” part of this document.

1. Post the Results to the Blackboard via “View/Complete Assignment” link.

The single deliverable file must be submitted via the above link for Assignment 1 in Blackboard. This will demonstrated in class as needed. Don’t forget how we do this, and the caution I mention, as we will be using the same method for the other assignments and the project.

### Assignment Deliverables:

1. A Microsoft Word 2007 Document containing your write up and findings. The name of this file will be **CPTN230\_A1\_Analysis\_*lastname*.docx** where *lastname* is your actual last name. The format of the document is as follows. Do not deviate.

* Title page
  + Document name
  + Author
  + Creation Date
  + Course Number
* Table of Contents
* Introduction
* Operating System Information
* How to use your C++ software development environment including
  + Identification
  + Entering code from scratch
  + Importing code from another source
  + Compiling the application source modules
  + Linking the application object modules
  + Running the application executable program
  + Saving the application
* A one paragraph description of what the application does
* A description of the program variables and objects
* A detailed (line by line) analysis of all source code and header files including
  + The purpose of each non blank line
  + A description of the syntax of each line
  + A description of how the variables and objects “change” as a result of executing the line of code
* Conclusions about the application
* References documented using APA format

Each of these different sections should have a section or page break as needed. The TOC should be auto generated from the Section Headings. The References page should be auto generated from the in-line citations. This is easy if you are using Microsoft Word 2007, which is now the industry norm. Using other compatible products will only cause you more work.

One of the goals of all my courses is to get folk used to creating professional looking documents even when they are technical in nature. This may seem like a pain now but if you incorporate the practice as you write documentation it becomes easy and second nature.

Note: the file must have the .docx file extension and open by default with Microsoft Word 2007, and now 2010.

### Strenuously Recommended Approach:

Even though the word “recommended” is in this section header, consider it as mandatory. The approach below includes requirements and “how to” suggestions that should be used to complete the assignment. Hopefully by following the steps in order and using the suggestions, completing the assignment won’t be a difficult task.

1. Understand the problem.

Before you begin any software development task you must fully understand what is required to complete the task. How else can you know when you are done? Here are some important things to do to complete this step.

Completely read this document from beginning to end before doing anything else. The various sections of this document interrelate. Questions raised in one section are often answered in another section. This will give you the big picture of what is going on.

Ask me questions for clarity but they should be specific questions as they relate to the assignment requirements or procedures. A question like, “How do I do this?” is not a specific question. Asking questions like “Can I do the assignment another way?”, “Can I use a different documentation application or submission process?”, or “Can I turn it in late?” also won’t work. The answer to these questions will be “No.” Part of this assignment is to follow an organization’s formal procedures and policies, use the required tools and manage your time just as you would have to in industry.

Now to be realistic. I know you and I cannot think of or cover every issue or question that may come up. Please feel free to post questions to the FAQ. I check Blackboard every day.

I would even check the FAQ forum first in case your question has already been asked and answered.

1. Identify and gather resources.

Under the “References” section, I listed some of the references you might use for this and future assignments. In reality, you may use any resource or reference you want except other people’s work. I am the person you should come to for help. Our textbook actually contains all the information on everything I am asking you to do but use other technical resources as desired.

You should already know how to use Blackboard, Microsoft Word and a C++ software development environment. If not, learn quick! We are not using any advanced features from any of these tools. You may if you wish, but don’t waste time learning and trying things you don’t need for this course.

A critical resources to you are the 7 files associated with this assignment. I recommend you print out hard copies of this document and the three .pdf files. In total, they require around 20 pages of printed material. I find it much easier to work from hard copies than from electronic copies. It is up to you.

1. Start your documentation.

“Is he serious? I haven’t done anything yet to document.” Yes, I am serious about doing this step now.

I suggest you start your Word document with the headings already in place, even thought there is no content yet. So how do I get started? Well here is an idea.

Since you should already have this document down loaded to your system from the previous step, why not make a copy of it with the correct file name. Then remove all content except the section headings and rename the headings to match the required outline. When you have the new headings only file, make a second copy under a generic name for future use. A quick, easy approach to get you started and have something you will be able to use over again for future assignments.

It should have the Title page, Section Headings, Auto TOC and a References page. Save it away (remember where you saved it) because you will need it for all of our assignments.

And yes, you can actually write the first bit of documentation. List the operating system type and C++ software development environment you are using in the appropriate sections.

1. Add the document introduction.

One paragraph should be enough. You can use this document as a source for this purpose. But just don’t copy and paste.

1. Compile and link the .cpp and .h files to create the application.

Using the system and C++ software development environment of choice (I suggest the ones supplied by NCC) compile and link the supplied code until it successfully creates a usable executable program. As you are doing this step I suggest you document the detailed step by step sequence. If you do it as you go it is much faster that going back and writing or recreating it from notes. In all cases, documentation should be created and updated as you go instead as a separate step. Trust me, it makes the process faster and life a lot easier.

There are three files supplied that you will need to create the application. Do not modify these files. They are:

* CPTN230\_A1\_applicatoin.cpp – application source code
* CPTN230\_A1\_class.h – class header file
* CPTN230\_A1\_class.cpp – class source code

Depending upon which C++ software development environment (also referred to as SDE in this document) you are using things will be different but the general sequence is something like the following. Adjust as necessary.

* Create and change to an empty folder
* Open the SDE
* Create whatever project/solution item needed by the SDE. This should create an initial set of project/solution files
* Import and open the application source file
* Compile only this file. The application cannot be linked at this time. This should initially populate the project/solution files. You should expect errors at this point. Some SDEs do not allow a separation of the compilation and link events
* Import and open the class header file and compile the application code again. This should update the project/solution files
* Import and open the class source file and compile the class code. This should update the project/solution files
* Build the entire project/solution. This includes compiling both source files and linking them together. The result should be a usable executable program
* Run the application once to verify it runs
* Save any files that need saving
* Close the C++ software development environment

1. Document the procedures used to create and run the application.

If not already done, document the procedure you used in the previous step. I hope you took my suggestion and did it as you went. If not, you get to repeat previous sequence to ensure you did not miss any steps in your documentation.

1. Run the application and see what it does.

In case you had not noticed to this point, there has been no information given on what the application does. If you have looked at the source files, there are no comments other than file name, author and date. The only clues as to what the application does are possibly in the function and variable names used. This is deliberate. It is your task to figure out what it does and how it does it.

Run the application multiple times and as you do take notes. Try using different sets of numbers. Vary the values (large, small, negative) of the numbers used. Note the differences in the output from the application. Try to break it by not following the on screen directions.

Before you can do a detailed analysis of the code you need to have a general idea of what the application does, so this step is important.

1. Write and overview of what the application does.

From your notes, write a paragraph (or two) describing what the application does. This overview is about what the application does and not how it does it. View the source files if you think it will help in your understanding, but base the overview on your direct observations from when running the application. Note differences in output, based on differences on input. Note any failure scenarios you see. Add this overview to your document in the appropriate section.

1. Develop a big picture of the contents of the source files and how they relate.

This is the point where you will probably want to switch over to using the .pdf files that contain exact copies of the supplied .cpp and .h files. These were supplied for one primary reason. They contain line numbers that should make it much easier to follow and document the code.

**Warning:** I noticed on line 74 in the application source file in the pdf copy an underscore character was overwritten. I can’t fix this. If there is any doubt as the line listed in the pdf file see the actual source or header file for clarification.

Again the purpose here is to figure out what is in each file and not how the application works. For each file I would list what functions, member functions, variables, data members and objects are defined/declared/prototyped or exist. Write this information down (with line numbers) so you have a “map” of the code in the applications files. This is the big picture. You will not be asked to submit these notes, this is just a tool to help you keep things in focus.

It is worth mentioning here that your analysis is only of the code in the three files. Do not include any information about system supplied files.

1. Start with an analysis of the class header file.

In all your analyses please do not list any comments or blank lines in the files. In the class header file, I know lines 1 - 7 are comments. I know line 8 is a blank line.

For the class header I want to know things like what the “#include”, “using” and “#define” statements are doing. Why did I use or need them?

In the actual class header file, what are the data members, what are the member functions, why are some things private and some things public? Are there any member functions provided that are not explicitly listed in the header file? Add all this information to your document.

1. Starting from the application analyze the code and document everything that is being done.

This step is the heart of the assignment. All the other requirements are important but this is the most important. It will demonstrate your skills at reading the source code for an existing C++ application. The code uses techniques covered in CPTN175 so you should be familiar with all syntax and techniques. If not, I need to know this.

But before you start here are some guidelines to follow when doing the analysis.

You need to decide on a documentation method. The two prime choices are pseudo code or flow charts. The choice is up to you but consider the following.

I prefer and suggest pseudo code. Pseudo code is a semi formal text method to describe real code. It is easy to create and most SDLC design documents use it. It is simple text and can be quickly transferred to a source code file and made into the comments for the source code. In this case we are reverse engineering the source code back to a design.

The other documentation method is the use of flow charts. You may be familiar with the technique and prefer to use it but there are some issues. First, a flow chart is a pictorial representation of code. If you want to use it in a Word document you will have to create objects and text in the document. This can be a laborious, time consuming process. The alternative is to create the flow chart with another tool and import the picture object into the Word document. Remember the assignment deliverable is a single Word document with all the information. Second, if you do design and document code with flow charts, when it comes time to write the actual code you will still have to manually enter all the comments in to the source file.

As mentioned before, do not document comments and blank lines in the source files. I know they are there. But you might want to mention why they are there and the impact they have on the source files in your conclusions.

Avoid redundancy in the analysis. You will encounter cyclic (loops) code. You do not need nor want to document the execution of the loop for every cycle. Analyze the code in the loop once and document what makes/breaks the loop. Within the various files you will see the same thing/method/algorithm used multiple times. You only need to provide an analysis of the first time you encounter the “thing.” When you see it used again, just note it is the same “thing” as the first time you saw it and provide a reference. A simple example is the repeated use of the same “#include” statements in multiple files. If a function or member function is called more than once you only need to document it once.

Any other code analysis issues or questions not addressed by the rest of this step should be directed to me.

One last thing. As you are doing the analyses if you cannot provide the requested information don’t get hung up. Skip that part and move on to the next piece of code. Often seeing what a later piece of code does helps clarify what happened in an earlier piece of code. Better an accurate but incomplete analysis than no analysis at all.

Now on to the real fun…

Make sure you have completed the previous steps before starting this part. It will make life a lot easier and save you a lot of time. If you have not and you ask questions covered in the previous steps, I’ll just direct you to complete them.

Start with the application source file and list (in your document) any global variables and objects as well as any local variables and objects in the main function. For any global or local main variables or objects list their type, purpose and initial values. For any objects list the data members and their type, purpose and initial values when the object is created. See your class header file description above. Also list any function prototypes listed in the application source file. Use line numbers in your analysis to identify to which code you are referring. You do not need to retype the code in the analysis, that is wasted work since I also have copies of the .pdf files. Add all this information to your document in the appropriate sections.

The actual analysis I’ll leave to you but here are some things to do for various lines of code. I also have some specific questions or things for you do add to the analysis.

Now start at the beginning of the main function. Line 18 in the application source file and proceed on to the code that does something. Remember you have already documented all main function local variables and objects.

For each line of code provide a simple statement of what it does plus any additional details to fully describe the actions of the statement. When any line of code modifies a variable or data member document the change that occurs to that variable or data member. Update the document as you go.

When analyzing a line of code that calls a function include the following information. The term function here also refers to calls to class member functions. When the function is activated where (by file and line number) is control transferred and when the function returns, to where (by file and line number) is control returned. Update the document. Documenting a function should only be done once and treated as a separate block of code. Do not include one function code analysis inside another function code analysis.

When analyzing a function show what local variables or objects are created. What are their types, purpose and initial values. Make sure to include the local variables or objects created as parameters to the function and add the method of passing the arguments. Were they pass by value or pass by reference? What happens to the functions local variable’s when the function returns? Are there function success/fail status codes involved? If so what are they and (if they are checked) how are they used? Update the document for each function called.

Some last things to address as you encounter them in your analysis and add to your document.

On line 33 in the application what is the purpose of the ‘\’ characters?

In the lines that use the “cout” object, sometimes I use flush, endl or ‘\n’ as part of the statement. What is the purpose of each and why choose one oven another? In these code statements sometimes I use the ‘’ (single quotes) instead of the “” (double quotes) to quote things. What is the difference? What is the purpose of the << operator in these statements and what is it actually doing? Answer these questions only once and add the answers to the document for the line in which you first encountered them.

When calling a member function sometimes I used the ‘.’ (dot) operator and sometimes I used the ‘->’ operator. What is the difference and why did I use them?

Throughout the code I used a mixture of return and exit statements. What is the difference? Why did I use different numbers for them? What do the numbers represent and how are they used? Some of the return statements don’t have any numbers. Why?

And lastly for this step, line 69 in the application source file is a return statement. What hidden event happens here?

Remember if you cannot answer a question or do part of the analysis because you don’t know or can’t research it, skip it. Don’t let one thing be an excuse to give up and stop work. Get as much done as possible. Partial work counts as long as it is correct.

1. Document your application conclusions.

The code used to build this application is not perfect. In fact, far from it. I want your conclusion of the techniques used in the code for this application. Specifically, consider addressing the following issues.

* Inconsistent use of techniques
* Unnecessary use of techniques
* Scope of variables and objects
* Error checking methods
* Are there ways to make the code safer/better

This information is often what you would be expected to present at a code design review meeting using your analysis to back up your conclusions. For this assignment there is no presentation required but your written conclusions are a requirement.

Add these conclusions to your document in the appropriate section.

1. Build the references page from the in-line citations.

I can think of at least three references you will be using. There may be more…

* The text book
* The system
* Blackboard
* Possibly Internet FAQ sites

1. Proof read your document.

I am not kidding. This document represents you, your skills and your professionalism in communications. In industry, people (especially the non-technical) notice the smallest details. Here are things to check.

* Information presented must be accurate and as complete as possible
* Spelling – I better not see any red squiggles when I turn on the spell checker other than for technical terminology
* Grammar – I should see a minimum of green and blue squiggles when I turn on the grammar checker
* Document formatting
  + Bulleted list items normally don’t end with periods
  + Numbered list items do end with periods
  + Consistent use of heading types
  + Consistent use of fonts, font sizes and styles for text
  + Consistent use of indenting of all text and lists

Yes, I will be checking and points deducted if there are too many issues.

And yes, we are not perfect. Things do slip by. You might even find some of these issues in this document, but hopefully not too many. If you do please feel free to let me know. I like my documents to be accurate and professional as well.

1. Rebuild the TOC.

This is essential as things may have moved around in the document or you may have added or deleted section headings.

1. Post the Word document to Blackboard via the “View/Complete Assignment” link.

Use the above link for assignment submissions. For those not familiar with this Blackboard method of assignment submission, it will be demonstrated in class. Do not submit assignments via the Blackboard Digital Drop Box or forums. Do not use e-mail attachments. Do not hand in hard copies of documents. This is the only acceptable submission method.

Remember time management is also an assignment requirement. Even if you have not completed the assignment you must submit what you have on the day the assignment is due. The due date will be announced in class. Partial credit is given for partially completed assignments as long as the completed parts are correct. No late submissions will be accepted. In industry when your manager assigns a task and wants something to be done, they expect it on time. It is part of the “salary continuation plan.”

1. Down load and test the deliverable.

It is your responsibility to make sure the file was uploaded and can be read. The only way I know to test this is to down load a temporary copy and make sure you can read it. Submission of unreadable files is not a valid reason for missing an assignment.

### A few Last Thoughts:

I know this assignment write up has been rather lengthy. Future assignments won’t be quite as verbose as this one. With this write up I tried to accomplish several goals. These include but are not limited to:

* Provide explicit, detailed and clear directions and requirements for the assignment
* Provide useful hints and techniques that you can use and reuse in future assignments
* Provide a rational for why we are doing things the way we are
* Share some of my observations about why and how I do things when it comes to software development
* Address issues and answer questions before they cause work blockage
* Set the expectations for future software development assignments and projects with methods that can save you development time

I also want to remind you that the primary purpose of this course is C++ Programming. All the other stuff will take care of itself after you get used to doing it as part of completing the assignments.

The requirements for this assignment may seem heavy handed and totally inflexible. If you think this then you are correct. But…

These requirements are still less rigid than those you are likely to encounter in industry. One of the goals of this course and the entire NCC program is to get you ready to work professionally in a professional environment. Remember your work represents you to your managers, fellow workers and eventual employees. Professionalism gets you a long way. Anything else gets you ignored and left behind.