**Seneca logo**[**CPR101**](https://cpr101.ca/index.html) **— Final Project**

|  |  |  |
| --- | --- | --- |
| 1 | [CP4P\_IT Projects and Project Management](https://cpr101.ca/FinalProject/CP4P_IT%20Projects%20and%20Project%20Management.pptx) | Project Management PowerPoint slides |
| 2 | [Final Project Overview](https://cpr101.ca/FinalProject/Final%20Project%20Overview.pdf)  READ THIS FIRST  Before class | ...just what it says |
| 3 | [Bb group - MS team set up](https://cpr101.ca/FinalProject/Final%20Project%20group-team%20set%20up.html)  Groups will be the presentation groups Set up groups in MS Teams (Include the Professor in your group)  If you have problems setting up in MS Teams you can use Google Docs and invite the professor.  [**Project Milestones and Details**](https://cpr101.ca/FinalProject/ProjectDetails.html) | Project Launch notes  **Milestone deadlines** |
| 4 | [Project Management notes](https://cpr101.ca/FinalProject/Project%20Management%20notes.pdf) reference [Final-Project-Plan](https://cpr101.ca/FinalProject/Final-Project-Plan.xlsx) template | Project Management Planning |
| 5 | [Programming Comments](https://cpr101.ca/FinalProject/Programming%20Comments.pdf) requirements [Programming Comments DEMO.mp4](https://cpr101.ca/FinalProject/Programming%20Comments%20DEMO.mp4) | Commenting Source Code |
| 6 | [Programming Test Cases](https://cpr101.ca/FinalProject/Programming%20Test%20Cases.pdf) requirements [Final-Project-Test-Cases](https://cpr101.ca/FinalProject/Final-Project-Test-Cases.xlsx) template for submission [module-test-cases](https://cpr101.ca/FinalProject/module-test-cases.xlsx) template for development | Creating Test Cases |
| 7 | [Compiling modules](https://cpr101.ca/FinalProject/Compiling%20modules.pdf)  [Minimalist gcc](https://cpr101.ca/FinalProject/Minimalist%20gcc.pdf)   * [**cl** compiler](https://youtu.be/rqLbyj0TnIg) video * [**gcc** compiler on Windows](https://youtu.be/DaiJoWr5fH0) video * [**gcc**compiler on macOS](https://youtu.be/we2Oc4WQ7FM) video | Dealing with C modules  command line compilers: gcc and MS cl |
| 8 | [git basics](https://cpr101.ca/FinalProject/git%20basics.pdf)   * [git Installation](https://youtu.be/Rhc0KzfLaBk) video * [git Essentials](https://youtu.be/mNnIh05iKjo) video | git version control for Project Ver.2 & 3 |
| 9 | [Project Source Code.zip](https://cpr101.ca/FinalProject/Project%20Source%20Code.zip) (archive) | Source code images and templates |
| 10 | [Teamwork](https://cpr101.ca/FinalProject/Teamwork.pdf) [THE PLAN](https://cpr101.ca/FinalProject/THE%20PLAN.pdf) | additional explanatory notes |
|  | **Deliverables with due dates Submitted in Blackboard**  **NO ZIP Files - Only zip file excepted will be for the Project Source Code** |  |
| 11 | Project Plan (In Excel) and Meeting notes (word Doc) At least two meetings  **Sunday July 30 @23:59** | Project Plan (Excel)  Minutes of Meetings (MS Word) |
| 12 | V#1 - New Updated Project Plan (In Excel) as of submission and Meeting notes (Word Doc) At least two meetings  Test Cases – module test case  Zip File of Source Code  Word Document of a summary of work completed and by which member(s) of the group.  **Sunday August 6 @23:59** | Updated Project Plan (Excel)  Minutes of Meetings since last week (MS Word)  Test Cases (MS Word)  Zip File of Source Code |
| 13 | V#2 - New Updated Project Plan (In Excel) as of submission and Meeting notes (word Doc) At least two meetings  Updated Test Cases – module test case  Zip File of Source Code  Word Document of a summary of work completed and by which member(s) of the group.  **Wednesday August 9 @23:59** | Updated Project Plan (Excel)  Minutes of Meetings since last week (MS Word)  Test Cases (MS Word)  Zip File of Source Code |
| 14 | V#3 - New Updated Project Plan (In Excel) as of submission and Meeting notes (Word Doc) At least two meetings  Updated Test Cases – module test case  Zip File of Source Code  Word Document of a summary of work completed and by which member(s) of the group.  ***Sunday August 13 @23:59 before the start of FINAL EXAMS*** | Updated Project Plan (Excel)  Minutes of Meetings since last week (MS Word)  Test Cases (MS Word)  Zip File of Source Code  Word Document of a summary of work completed by group member which includes the names of all members of the group |

**Notes**

**See your Blackboard course for project specifics, milestones (due dates), and details that may differ or not be available here.**

**Most people have negative experiences with academic group work** because project managment is done more ad hoc ("to this") than by agenda ("to drive on, set in motion").

Doing any project by gut feel, with the best of intentions, according to "Why waste time planning when there is so much to do?", "If it's not on fire, it's not important." is adequate for small tasks and things that can wait like eating, sleeping, hygiene, and other people. With proper neglect, the second-last will take care of the very last. If it is called a project, it is not a small task; projects requirement management.

**It *is* possible for academic group work to be successful and enjoyable.** The investment up front in explicit project management pays dividends quickly. Doing the management is a marked component of this project. It's not just overhead. We pay you to do it. (Industry does not pay Project Managers six figure salaries to do busywork and waste other people's time.)

**There is indeed a lot of documentation here.** Why? Because it answers all the who, what, when, where, how, and why questions students have previously asked about similar projects. Because it is the level of documentation you will see at the implementation stage of professional projects. Everything about the project arrives all at once because the senior project manager (your prof) expects each programmer (you) to be their own personal project manager. Those who do not project manage themselves are coders, not programmers. The SDDS CPP & CPA program is for programmers.

Any project has far too much documentation to be absorbed all at once whether one is a first term student or a seasoned professional.   
Projects are managed in progressive stages so Everything does not happen Everywhere All at Once.   
How do you eat an elephant? One bite at a time.

[Real-Life Problems of Programmers](https://blog.devgenius.io/10-hilarious-cartoons-that-depict-real-life-problems-of-programmers-36b2287679e8)