# CIS11 Course Project: Building Win32 Applications Using MASM

This project is designed to capture primary concepts of the course, which consist of computer architecture, processing and assembly programming. For this project, you will be working in teams, maximum 3 students per team. Each team will plan, design and program Windows x86 applications. Refer to below requirement list for Course Project.

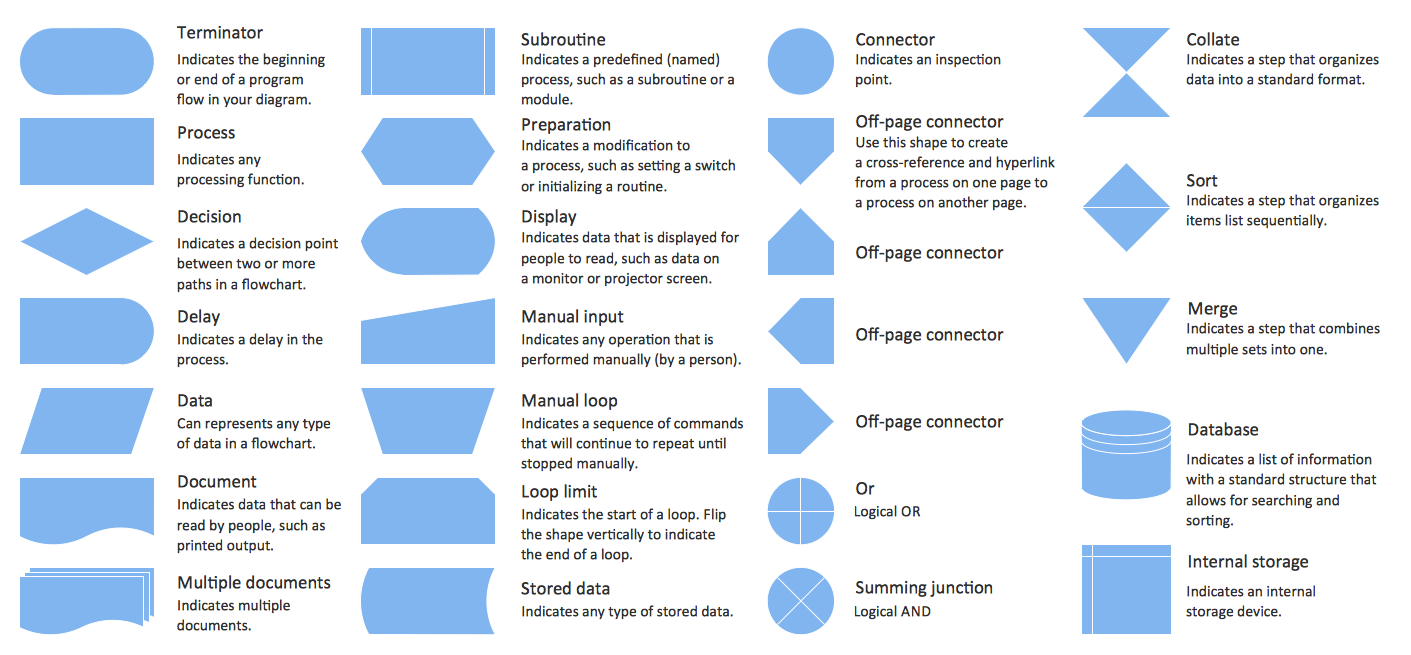
1. Form a team and select team name.
2. Visit the following websites for ideas and information to build programs in Windows x86. Each group will be required to build 3 programs in assembly language, MASM, using MASM32 SDK.

* Win32 application examples: <http://win32assembly.programminghorizon.com/tutorials.html>
* MASM32 example: <http://www.masm32.com/teaching/>
* MASM Windows: <http://www.dreamincode.net/forums/topic/252104-masm-our-first-window/>
* MASM and Win32 Part 1: <https://0x62626262.wordpress.com/2015/10/10/win32-assembly-part-1/>
* MASM and Win32 Part 2: <https://0x62626262.wordpress.com/2015/10/16/win32-assembly-part-2/>
* MASM and Win32 Part 3: <https://0x62626262.wordpress.com/tag/masm32/>
* Assembly Language Windows Programming: <https://www.bigmessowires.com/2015/10/06/assembly-language-windows-programming/>
* Win32 Programming for x86 Assembly Language Programmers: <http://www.oopweb.com/Assembly/Documents/Win32ASM/Volume/win32asm.htm>
* Win32 Assembly Part 2: <http://archive.gamedev.net/archive/reference/articles/article757.html>

1. Assembly programs must have the following requirements:
2. Include comments in the program.
3. Appropriately link to libraries.
4. Data and code are segmented.
5. Contain correct entry point and format.
6. Declare variables and arrays.
7. Utilize standard entry and exit sequence to stack frame.
8. Contain procedure calls, parameters, arguments, instructions and directives.
9. Use dynamic memory allocation and cannot have overflow.
10. Program must have interactive feature and control (input from mouse or keyboard).
11. Program must be built as Windows application (.exe).
12. Program must go through debugging process before release.
13. At least 1 program must contain conditional directives (if, elseif, else).

**Note**: Online examples and open-source resources can only be used as references. Exact copy-paste of code is NOT permitted.

1. Create a flowchart for each program. Refer to below picture for standard shapes. Use MS Word or diagraming tool to create the flowchart.



1. Use GitHub to document and manage program versions.
2. Visit <https://github.com/> to create user account. Team members must have access to individual GitHub account.
3. Click on the Read the Guide button (green) to see detailed instructions.
4. Each team will need to create **1 REPO**, see <https://help.github.com/articles/create-a-repo/> for instructions.
5. In the Repo, make a Commit change by editing the README file to include the following information.
6. Team Name
7. Project Information: MASM32 Win32 Application Project
8. Note **each application** details:
9. What is the primary purpose of the application?
10. What is its scope (goals and deliverables)?
11. What are the restrictions or constraints in the program?
12. What are the dependencies and assumptions of the program?
13. What are the technical functionality in the program? Refer to flowchart for details for each program function (input, output, process, data…)
14. From the main team Repo, set up fork clones for members. See <https://help.github.com/articles/fork-a-repo/>
15. From the main team Repo, create a branch for each team member. Branches are used to propose changes to GitHub projects. You must give write access to team members to access the main repository to create a branch and open a pull request. A branch is a parallel version of the main line of development in the repository, or the default branch (usually master). Use branches to develop features, fix bugs and safely experiment with new ideas.

See <https://help.github.com/articles/creating-and-deleting-branches-within-your-repository/>

1. When your team begin to modify code and make changes to assembly projects, create a Pull Request. See <https://help.github.com/articles/creating-a-pull-request/>
2. Each team member will create Gist to share single files, parts of files, and full applications with other members. See <https://help.github.com/articles/creating-gists/> and make a Public Gist for each program.
3. In team Repo, add members as Collaborators (Settings 🡪 Collaborators).

Submission:

1. Flowchart document for 3 programs.
2. GitHub Repo URL.
3. .exe file for each program.