**Assignment 2**

**Git Commands:**

**$ git init** - This command is used to initialize the repository

**$ git clone** - This command is used to make a copy of an existing repository to a local folder to work on it or make changes to it.

**$ git status** - This command will give the status of changes that are not staged and staged for commit.

**$ git add <*filename>*** - This command will add the unstaged file to staged for commit.

**$ git add .** - This command will add multiple unstaged files to staged for commit.

**$ git commit -m “*message*”** - This command will commit/ save the staged changes. Writing a message everytime we commit is a good way to tell others our reason to commit.

**$ git push origin master** - This command pushes the committed changes to master

**$ git checkout -b <*master/branchname>*** - This command will switch to a working tree of respective branch specified

**$ git branch** - This command will provide which is the current working tree

**$ git checkout master** - This command will switch to master

**$ git pull origin master** - This command will fetch and download the latest modified repository to local

**How are high level programs gets converted to low-level machine code(binary code)?**

When we run a program, the program is sent into a compiler where the compiler converts our program into machine understandable language. By doing this, the compiler bridges the gap between high level program to low level machine code. The low level machine code is written in binary code. The code is first converted into tokens and these tokens are organized into hierarchical architecture which is a parse tree. Semantic analysis is performed to find scope and followed by some intermediate steps, it is converted into binary code.

**How are instructions fed to CPU?**

The binary code is a pattern of binary digits (0’s & 1’s representing ‘off’ and ‘on’) known as bits to each character. CPU is designed to read instructions memory and execute those instructions like add or jump instructions. CPU has an instruction processing unit and execution unit. The processing unit decodes the instructions to determine what the instruction is saying. The result of the instruction is stored into memory which might be the address of the next instruction.

**What is compiler?**

A compiler is also a program that converts a program into machine readable language. The compiler creates tokens, parses them, contextualizes and generates the object code. The object code is the machine code that the processor can execute one instruction at a time. A compiler is specific to one platform and when the program is executed in another platform then the compiler has to redo all the steps again to create binary code understandable by that machine.