PLAN 396
Lecture 1

Dr. Hossen Asiful Mustafa
https://hossenmustafa.buet.ac.bd

Reference Book

Herbert Schildt, Teach Yourself C (McGraw-Hill)

Dr. Mohammad Kaykobad, Dr. Md. Mostofa Akbar, Dr. M. A. Hakim Newton, Structured C/C Plus Plus Programming

David I. Schneider, An Introduction to Programming Using Python

Assessment

Type	Percent
Class Assignment	40
Midterm Quiz	10
Homework Assignment + Term Project	30
Final Quiz	20

Tentative Plan

- Introduction to Programming using C
  - 7 weeks
- Programming with Python
  - 6 weeks

The C Language

- Currently, the most commonly-used language for embedded systems
- “High-level assembly”
- Very portable: compilers exist for virtually every processor
- Easy-to-understand compilation
- Produces efficient code
- Fairly concise

C History

- Developed between 1969 and 1973 along with Unix
- Due mostly to Dennis Ritchie
- Designed for systems programming
  - Operating systems
  - Utility programs
  - Compilers
  - Filters
- Evolved from B, which evolved from Basic Combined Programming Language (BCPL)

C History

- Original machine (DEC PDP-11) was very small
  - 24K bytes of memory, 12K used for operating system
- Written when computers were big, capital equipment
  - Group would get one, develop new language, OS

C History

- Many language features designed to reduce memory
  - Forward declarations required for everything
  - Designed to work in one pass: must know everything
  - No function nesting
- PDP-11 was byte-addressed
  - Now standard
  - Meant BCPL’s word-based model was insufficient

Pieces of C

- Types and Variables
  - Definitions of data in memory
- Expressions
  - Arithmetic, logical, and assignment operators in an infix notation
- Statements
  - Sequences of conditional, iteration, and branching instructions
- Functions
  - Groups of statements and variables invoked recursively

C Types

- Basic types: char, int, float, and double
- Meant to match the processor's native types
  - Natural translation into assembly
  - Fundamentally nonportable
- Declaration syntax: string of specifiers followed by a declarator
- Declarator's notation matches that in an expression
- Access a symbol using its declarator and get the basic type back

C Type Examples

```
int i;
int *j, k;
unsigned char *ch;
float f[10];
char nextChar(int, char*);
int a[3][5][10];
int *func1(float);
int (*func2)(void);
```

C Compiler

- A computer can only read and execute binary or machine-code
- Compiler is a program that converts a C program into a machine-code (binary)
- We will use GCC (GNU C Compiler)

Integrated Development Environment (IDE)

- An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.
- An IDE normally consists of:
  - a source code editor
  - A build automation tools
  - a debugger
- An IDE supports intelligent code completion.
- We will use ‘Code::Blocks’ as the IDE http://www.codeblocks.org/

Hello World in C

```
#include <stdio.h>

void main()
{
  printf("Hello, world!\n");
}
```

Preprocessor used to share information among source files
- Clumsy
+ Cheaply implemented
+ Very flexible

Hello World in C

```
#include <stdio.h>

void main()
{
  printf("Hello, world!\n");
}
```

Preprocessor used to share information among source files
- Clumsy
+ Cheaply implemented
+ Very flexible

I/O performed by a library function: not included in the language

Program to Add Two Numbers

```
x = 5;
y = 10;
```

Program to Add Two Numbers

```
x = 5;
y = 10;
z = x + y;
```

Program to Add Two Numbers

```
int x, y, z;
x = 5;
y = 10;
z = x + y;
```

Program to Add Two Numbers

```
int x, y, z;
x = 5;
y = 10;
z = x + y;
printf ("The sum is %d", z);
```

Program to Add Two Numbers

```
#include<stdio.h>
void main(){
        int x, y, z;
        x = 5;
        y = 10;
        z = x + y;
        printf ("The sum is %d", z);
}
```

Class Assignment -1

- Install CodeBlocks
- Run the C program from the Lecture