

Computer Programming I

This is an introductory course to computer programming for engineers. Knowledge of C programming is commonly accepted as of primary importance for an engineer. This knowledge can easily be translated to any other programming language.

[Practical Exercise](#)

Course Syllabus

Week	Lecture	Laboratory	Notes
Week 01	• Introduction to EEC120 Computer Programming I • Introduction to Computer Systems 1. Hardware (CPU, input, output) 2. Software (Kernel, OS, userspace) 3. Numbers & Logic • Introduction to Computer Programming 1. Algorithms: instruction sequencing 2. Flow-charts: visual representation 3. Interpreters ⇒ scripts 4. Compiler and Linkers ⇒ binary executables	• Tools (compiler/IDE) Installation/Setup	• [VIDEO GUIDE]???

Week	Lecture	Laboratory	Notes
Week 02	- Introduction to C language 1. Basic code structure (syntax) 2. Function and code-block 3. Simple text display - Creating first executable binary 1. Syntax (language) and semantic (decision logic) errors 2. Simple debugging	- Explore building binary from c source code - Explore creating text decorations using symbols	[VIDEO GUIDE]???
Week 03	- C Variables (data storage) 1. data types and sizes - C Operators 1. arithmetic and logic 2. boolean vs bitwise - Data processing 1. equations and tasks 2. (optional) math.h library functions	- Understanding storage types and practical use - Exploring code that does equations/calculations	Assignment 1 Queue
Week 04	- Flow control: Branches (selection) 1. if-else structure 2. switch-case structure 3. (optional) the ?: combo	- Understanding decision making process in code - Exploring code that control program flow	
Week 05	- Flow control: Loops (repetition) 1. for structure 2. while structure 3. do-while structure	- Understanding iterative flow in code - Exploring code that repeats itself	Assignment 1 Due (5%)

Week	Lecture	Laboratory	Notes
Week 06	- C (Data) Array 1. storage in groups 2. accessing array items (content)	- Understanding array as group-storage - Exploring code that process on arrays (read/write)	Lab Assessment 1 (10%)
Week 07	- C String 1. array of chars (null terminated) 2. (optional) string.h library functions	- Understanding string as char array - Exploring code that process on strings (read/write)	Mid-term Examination (20%)
Week 08	- C Pointers 1. indirect data access (data that holds address to other data) 2. array name as address 3. pointers in loops 4. pointer arithmetic	- Understanding pointers as method for indirect access - Exploring code that deals with (or use) pointers	Assignment 2 Queue
Week 09	- C Functions 1. function return value 2. function parameters 3. (extra) data/variable scope and lifetime	- Understanding functions as building block for modular programming - Exploring code that has multiple functions	
Week 10	- File access (permanent storage) 1. accessing file on secondary memory (i.e. disks) 2. text vs binary file	- Understanding pros/cons of secondary storage - Exploring code that read/write from/to files	Assignment 2 Due (5%)

Week	Lecture	Laboratory	Notes
Week 11	- Practical Applications - requirements and considerations - C struct - data group - accessing data in struct	- Lab Project (extra) use struct to create abstract data type (extra) creating functions to process struct (towards oop?)	Lab Assessment 2 (10%)
Week 12	- Practical Applications (cont.) - requirements and considerations	Lab Project (cont.)	
Week 13	- Practical Applications (cont.) - requirements and considerations	Lab Project (cont.)	
Week 14	- Practical Applications (cont.) - requirements and considerations	Lab Project (cont.)	Lab Project Due (10%)

2026/08/11 07:42 · azman

Introduction to C Programming

Variable and Operators

[intsize.c](#)

```

#include<stdio.h>
int main(void)
{
    printf("Size of unsigned int: %d\n",sizeof(unsigned int));
    printf("Size of unsigned long long: %d\n",sizeof(unsigned long long));
    printf("Size of unsigned long: %d\n",sizeof(unsigned long));
    printf("Size of unsigned short: %d\n",sizeof(unsigned short));
    printf("Size of unsigned char: %d\n",sizeof(unsigned char));
    return 0;
}

```

[varsize.c](#)

```
#include<stdio.h>
int main(void)
{
    printf("Size of int: %d\n",sizeof(int));
    printf("Size of float: %d\n",sizeof(float));
    printf("Size of double: %d\n",sizeof(double));
    printf("Size of long double: %d\n",sizeof(long double));
    printf("Size of char: %d\n",sizeof(char));
    return 0;
}
```

From:

<https://www.azman.my1matrix.cc/> - Azman @MY1

Permanent link:

<https://www.azman.my1matrix.cc/doku.php?id=archive:eec120&rev=1786439982>

Last update: **2026/08/11 09:19**

