Short-term Hands-on Supplementary Course on C Programming

SESSION 1: C Programming Basics

NIVEDHITHA D
KARTHIK D

Time: 2:30 - 3:40 PM
Date: 12 May 2022
Location: Online

Agenda

1. Meet your instructors
2. Administrative Instructions
3. Goals of this course - Why C?
4. Structure of a C Program
5. Comments & Documentation
6. Keywords
7. Identifiers & Naming Conventions
8. Classification of Data Types
9. Variables
10. Constants
11. Basic I/O
12. Operators in C
13. Expressions
14. HANDS ON: Tutorial + Programming

Administrative Instructions

- Please fill out the feedback form - will be shared in the chat
- Join us on Microsoft Teams,

  Team Code: **rzlaicv**

GITHUB REPOSITORY! ⭐🌟

Meet the Team

[Dr. T.T. Mirnalinee]

Professor & HoD, CSE

[Dr. B. Prabavathy]

Associate Professor, CSE

[Nivedhitha D]

B.E. CSE (2018-2022)

[Karthik D]

B.E. CSE (2019-2023)

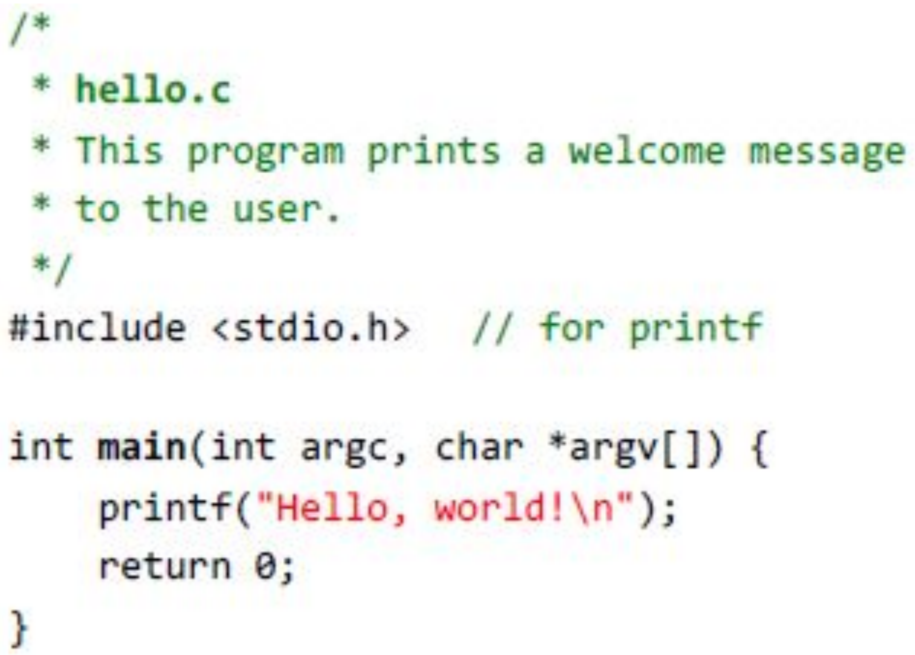
Goals of the Course

- Customized course to give you sufficient hands-on experience with the basics of programming
- C proficiency that will help you with your practical courses all the way until your 7th semester
- Hone your basics for the aptitude tests for placements
- Procedural programming paradigm based portfolio-ready capstone project
- Curated notes and roadmaps
- Bonus:
  - UNIX command line basics
  - GitHub account setup and basics
  - Latex Documentation
  - C Installation & Setup
  - Coding Parties
  - Ask Us Anything on the group related to the course, trivia and beyond!!

Agenda

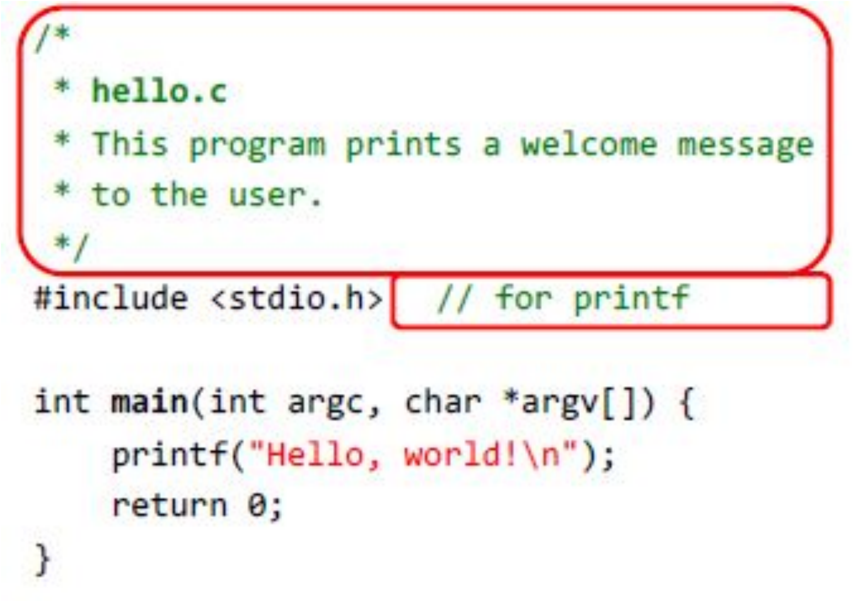
1. Meet your instructors
2. Goals of this course - Why C?
3. Structure of a C Program
4. Comments & Documentation
5. Keywords
6. Identifiers & Naming Conventions
7. Classification of Data Types
8. Variables
9. Constants
10. Basic I/O
11. Operators in C
12. Expressions
13. HANDS ON: Tutorial + Programming

Our First C Program



```c
/*
 * hello.c
 * This program prints a welcome message
 * to the user.
 */
#include <stdio.h>   // for printf

int main(int argc, char *argv[]) {
    printf("Hello, world!\n");
    return 0;
}
```

Structure of a C Program

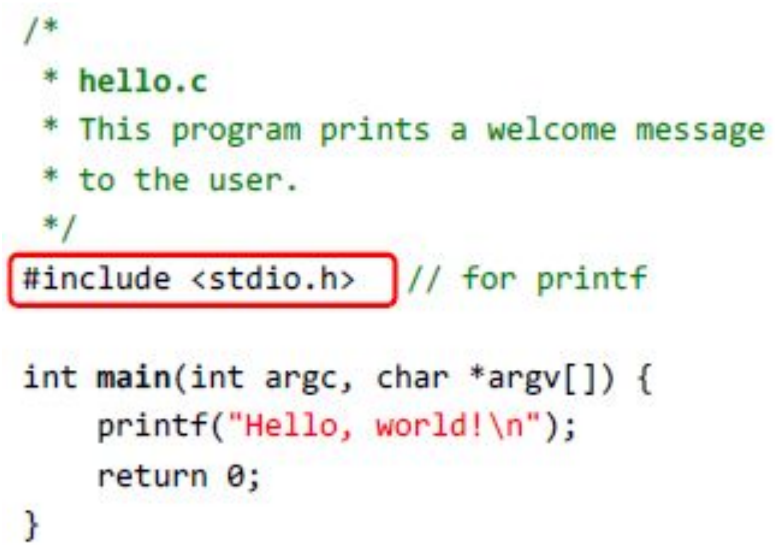


```
/*
 * hello.c
 * This program prints a welcome message
 * to the user.
 */
#include <stdio.h>   // for printf

int main(int argc, char *argv[]) {
    printf("Hello, world!\n");
    return 0;
}
```

[Comments & Documentation Style]()

Program comments
You can write block or inline comments.

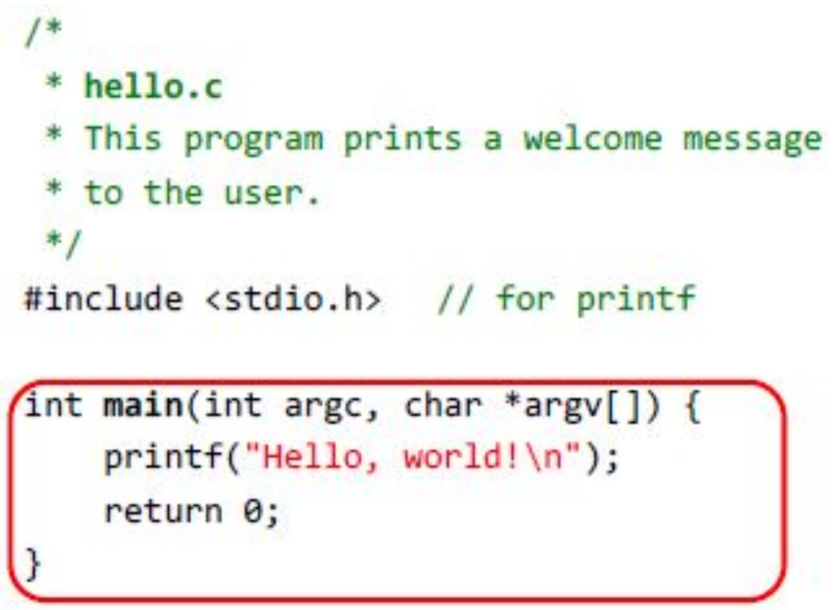
Structure of a C Program



```c
/*
 * hello.c
 * This program prints a welcome message
 * to the user.
 */
#include <stdio.h>  // for printf

int main(int argc, char *argv[]) {
    printf("Hello, world!\n");
    return 0;
}
```

Import statements
C libraries are written with angle brackets.
Local libraries have quotes:
`#include "lib.h"`

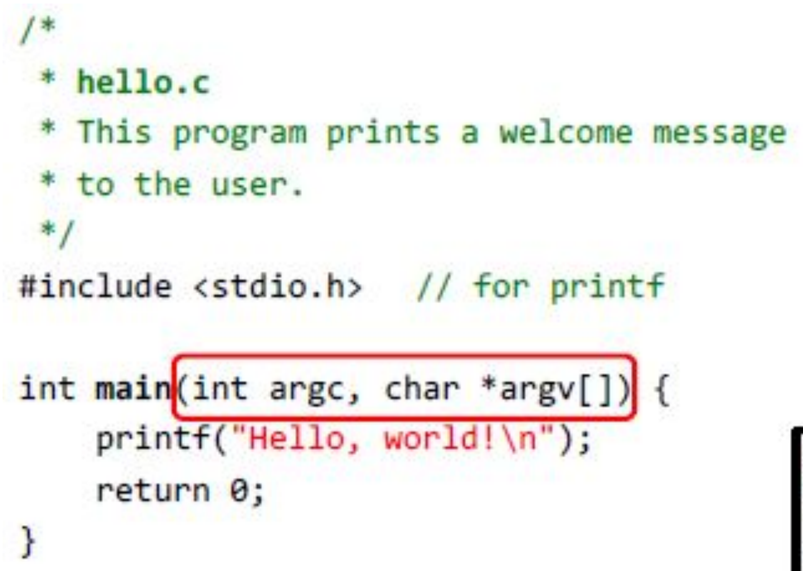
Structure of a C Program



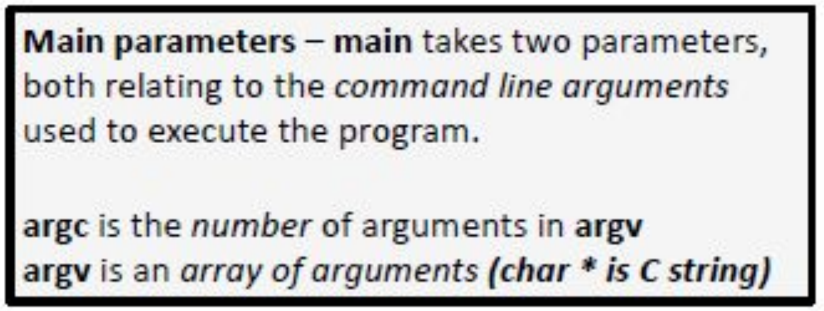
```c
/*
 * hello.c
 * This program prints a welcome message
 * to the user.
 */
#include <stdio.h>   // for printf

int main(int argc, char *argv[]) {
    printf("Hello, world!\n");
    return 0;
}
```

Main function – entry point for the program
Should always return an integer (0 = success)

Structure of a C Program



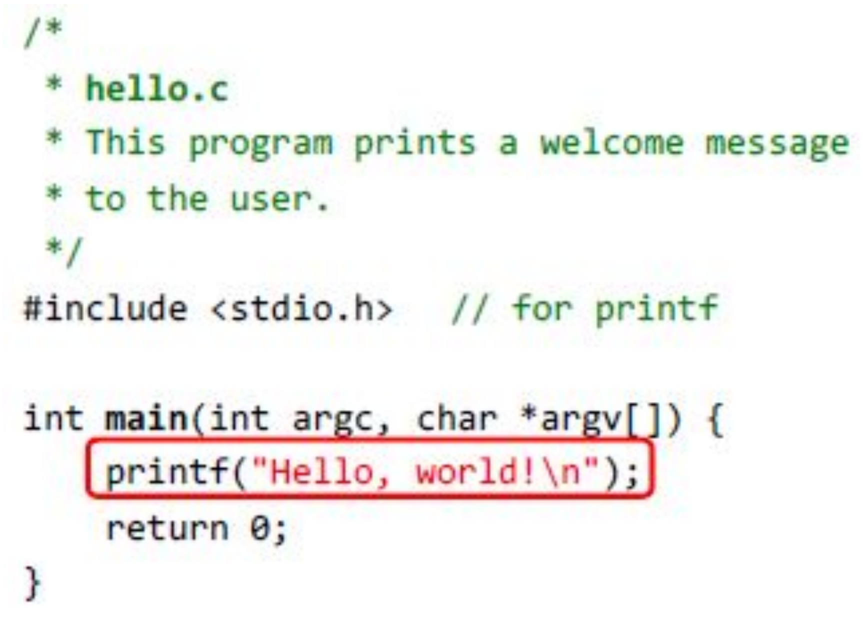
```
/*
 * hello.c
 * This program prints a welcome message
 * to the user.
 */
#include <stdio.h>   // for printf

int main(int argc, char *argv[]) {
    printf("Hello, world!\n");
    return 0;
}
```



Main parameters – **main** takes two parameters, both relating to the *command line arguments* used to execute the program.

argc is the *number* of arguments in **argv**
argv is an *array of arguments* ***(char * is C string)***

Structure of a C Program

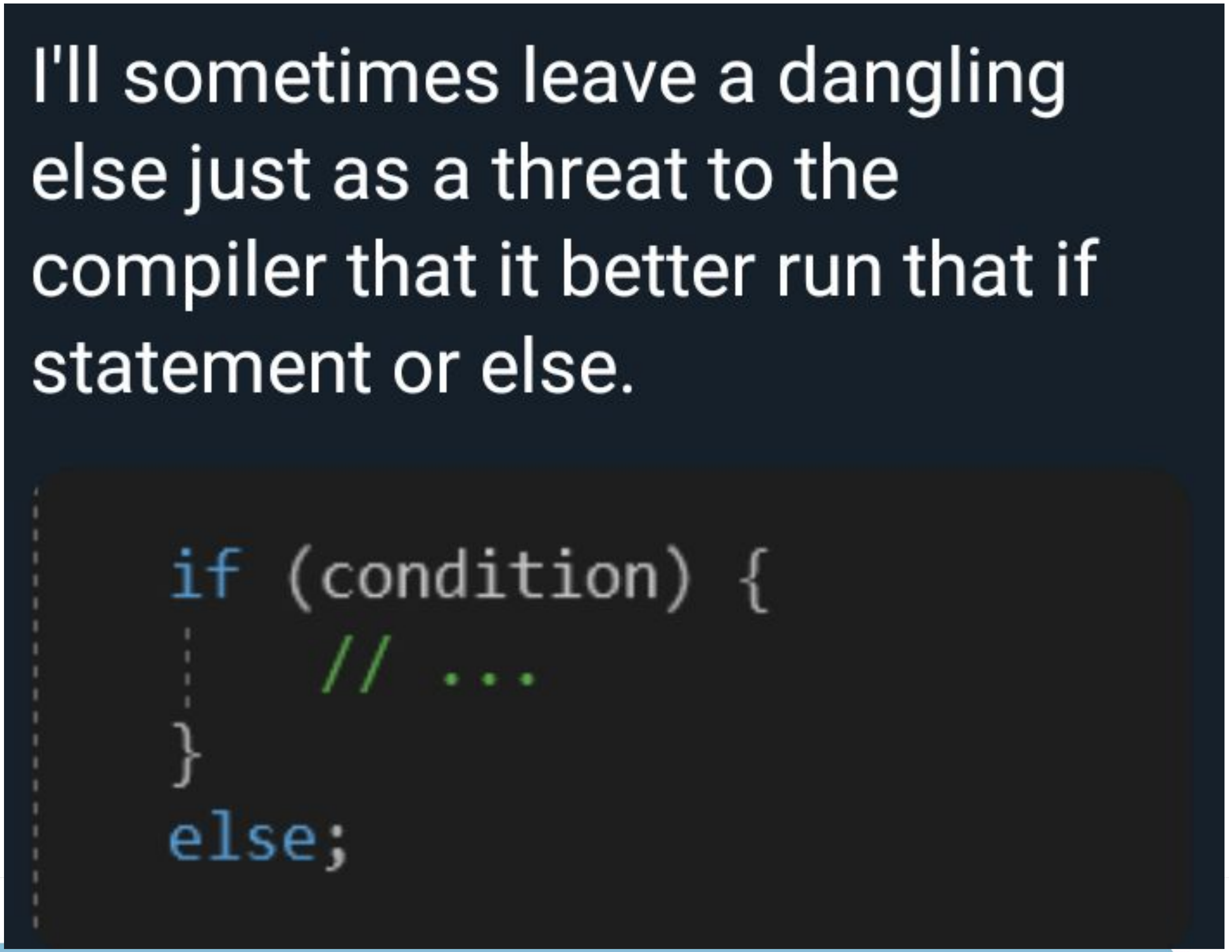


```c
/*
 * hello.c
 * This program prints a welcome message
 * to the user.
 */
#include <stdio.h>   // for printf

int main(int argc, char *argv[]) {
    printf("Hello, world!\n");
    return 0;
}
```

printf – prints output to the screen

Next Session

Flow of Control!!



I'll sometimes leave a dangling else just as a threat to the compiler that it better run that if statement or else.

```
if (condition) {
    // ...
}
else;
```

Any Questions