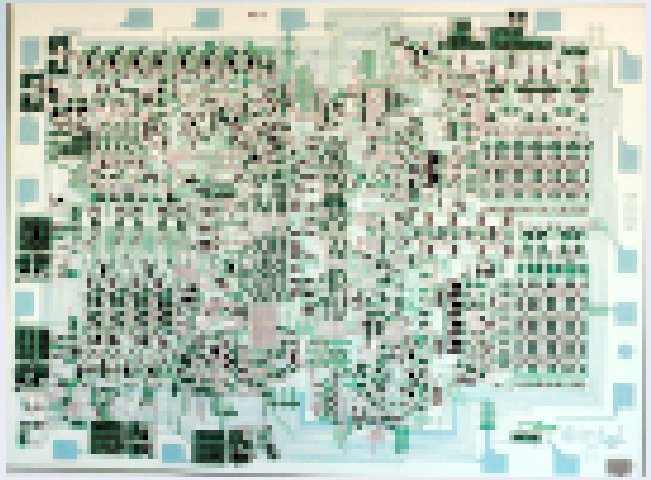

Lecture 12b: Programming in The C Language

CSIS11: **Computer Architecture and Organization**

Readings

- *Chapters 11-12* Patt and Patel: Introduction to Computing Systems...
- *The C Language* - Kernighan and Ritchie - Second Edition, strongly recommended
- Beej's Guide to C Programming (Tutorial) - Instructor Repo
- Beej's Guide to C Programming (Library Reference) - Instructor Repo

Quick Guide to Developing C Programs

- Rules for Success
- Template for C Program
- Variables
- Input and Output
- Control

Rules for Success

1. **End each statement with a semicolon (;)**
2. **Include necessary header files** (`#include <stdio.h>` for input/output functions)
3. **Every C program needs a main() function**
4. **Check your braces { }**, ensure opening and closing braces are properly paired
5. **Initialize variables before using them**
6. **Use proper data types**, match variable types with their intended use
7. **Add comments to explain your code**, use `//` for ALL comments

Template for a C Program

```c
#include <stdio.h>

// Function: main
// Description: counts down from user input to STOP
void main()
{
    // variable declarations

    // prompt user for input

    // define task

    // print message

}
```

Variables

- **Have to be declared** prior to using
- Keep scope in mind

```
// Integers
int myIntegerVariable

// Characters (not character string)
char myCharacterVariable

// Float
float myFloatVariable
```

Input

```
// simple input
scanf("%d\n", startPoint);

// multiple input, separate entries with a blank
scanf("%d %d", startPoint, counter);
```

Formatting options:

- `%d` - decimal integer
- `%c` - ASCII character
- `%f` - floating-point number

Output

```
// simple print
printf("%d\n", startPoint);

// print expressions, not just variables
printf("%d\n", startPoint - counter);

// print multiple expressions with a single statement
printf("%d %d %d\n", counter, startPoint, counter);
```

Formatting options:

- `%d` - decimal integer
- `%c` - ASCII character
- `%f` - floating-point number

Control Structure

In order to control the order of operation, use an *if-else if-else* construct.

```
if (comparison)
{
    code;
}
else if (comparison)
{
    code;
}
else
{
    code;
}
```