CSE 1101: Structured Programming Language

Lecture 13: Basic Input and Output

5.1 Input and Output in C

C provides various functions for input and output operations. The primary functions include:

- `printf()` for output
- `scanf()` for input
- `gets()` and `puts()` for string handling

5.1.1 Printing Output using `printf()`

The `printf()` function is used to display formatted output.

Syntax:
```
printf("format string", argument_list);
```

Example:
```
printf("Integer: %d, Float: %f, Character: %c\n", 10, 3.14, 'A');
```

5.1.2 Taking Input using `scanf()`

The `scanf()` function is used to take formatted input from the user.

Syntax:
```
scanf("format string", &variable);
```

Example:
```
#include <stdio.h>
int main() {
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);
    printf("You entered: %d\n", num);
    return 0;
}
```

5.1.3 Handling Strings with `gets()` and `puts()`

Example:
```
#include <stdio.h>
int main() {
    char name[50];
    printf("Enter your name: ");
```

Engr. Mohammad Mamun Hossain
BSc.(Eng.) and MSc.(Thesis) in CSE, SUST
PhD (Pursuing) in CSE, RUET.
Associate Professor, Dept. of CSE, BAUST

Contact: mhossain@baust.edu.bd
linkedIn:mamunsust12
Resources: mamunsust12@GitHub
Emergency: 01717690847

```
    gets(name);
    printf("Hello, ");
    puts(name);
    return 0;
}
```

Note: The `gets()` function is unsafe and should be replaced with `fgets()`.

5.1.4 Escape Sequences

Escape sequences are special characters represented with a backslash (`\`).

Escape Sequence	Meaning
`\n`	New line
`\t`	Tab
`\\`	Backslash
`\"`	Double quote

5.2 Header Files

5.2.1 Definition and Purpose

Header files contain function prototypes, macros, and definitions needed for a program. They are included at the beginning of a C program using the `#include` directive.

5.2.2 Types of Header Files

1. **Standard Header Files:** These are predefined and come with the C library.
   - `<stdio.h>`: Standard input/output functions (e.g., `printf()`, `scanf()`)
   - `<stdlib.h>`: Memory management functions (e.g., `malloc()`, `free()`)
   - `<math.h>`: Mathematical functions (e.g., `sqrt()`, `pow()`, `abs()`)
   - `<string.h>`: String manipulation functions (e.g., `strlen()`, `strcpy()`)
2. **User-Defined Header Files:** Created by programmers for modular programming.

5.2.3 Including Header Files

- Using Angle Brackets `< >` for standard files:

```
#include <stdio.h>
```

- Using Double Quotes `" "` for user-defined files:

```
#include "myheader.h"
```

5.2.4 Creating and Using a User-Defined Header File

Step 1: Create a Header File (myheader.h)

```
// myheader.h
#ifndef MYHEADER_H
#define MYHEADER_H
void greet();
#endif
```

Step 2: Create a Source File (myheader.c)

```
// myheader.c
#include <stdio.h>
#include "myheader.h"
void greet() {
    printf("Hello from my header file!\n");
}
```

Step 3: Use the Header File in Main Program

```
#include <stdio.h>
#include "myheader.h"
int main() {
    greet();
    return 0;
}
```

5.2.5 The `#define` Preprocessor Directive

The `#define` directive is used to define constants or macros.

```
#define PI 3.1416
printf("Value of PI: %f", PI);
```

Summary

- Data types define the type of data a variable can hold.
- The size and range of data types depend on system architecture.
- Basic I/O operations in C include `printf()` for output and `scanf()` for input.
- Header files store function declarations and macros, allowing code modularity.
- C has standard and user-defined header files for better code organization.

This lecture note provides foundational knowledge on **Data Types, Basic I/O, and Header Files**, ensuring students understand fundamental structured programming concepts in C.

Engr. Mohammad Mamun Hossain
BSc.(Eng.) and MSc.(Thesis) in CSE, SUST
PhD (Pursuing) in CSE, RUET.
Associate Professor, Dept. of CSE, BAUST

Contact: mhossain@baust.edu.bd
linkedIn:mamunsust12
Resources: mamunsust12@GitHub
Emergency: 01717690847