c_programming_session_1

اللهمَّ علِّمنا ما ينفعنا، وانفعنا بما علمتنا، وزدنا علمًا. وافتح علينا فتحًا عظيمً.

Tags: c programming
Status: #Adult

C Programming Session 1 - Fundamentals

Overview

This session covers the fundamentals of C programming as part of the NTI program, establishing a foundation for embedded systems development.

Instructor: Eng. Mohammed Abd El-Naeem (Mohammed Mega)

- Experience: Valeo, IoT startups, web applications

Embedded Systems vs General Purpose Computing

Feature	General Purpose Computer	Embedded System
Purpose	Multi-purpose, flexible	Specific, dedicated tasks
Examples	Desktop, laptop	Washing machine, refrigerator, fan controllers
Memory	Large, expandable	Limited, optimized
Connectivity	Various interfaces	IoT-enabled for specific functions

System Architecture Types

```
                    Configurable
System on Board (SOB) <-----------> System on Chip (SOC)
                                    └── AVR (memory + peripherals)
```

C vs C++

Feature	C	C++
Programming Paradigm	Structural programming	Object-Oriented Programming (OOP)
Complexity	Simpler, bit-level operations	More complex, advanced features
Memory Control	Direct bit manipulation	Higher-level abstractions
Use Case	Embedded systems, low-level	Applications, systems programming

Variables and Data Types

Primitive Data Types

Type	Size	Description
`char`	1 byte	Character/small integer
`int`	2-4 bytes	Integer numbers
`float`	4 bytes	Floating-point numbers
`double`	8 bytes	Double-precision floating-point

Variable Naming Rules

- Characters: a-z, A-Z allowed
- Numbers: 0-9 allowed (not as first character)
- Special: Only underscore (_) allowed
- Start: Must begin with letter or underscore
- Spaces: Not allowed
- Uniqueness: No duplicate names
- Reserved: Cannot use system keywords

Format Specifiers

Specifier	Data Type
`%c`	Character
`%d`	Signed integer
`%e`	Scintific notation floats
`%f`	Float
`%g`	Float with current precision
`%s`	String
`%u`	Unsigned integer
`%x`	Hexadecimal
`%%`	Literal %

Operators

Arithmetic Operators

```
// Binary operators
+ - * / %

// Unary operators
++x  // Prefix increment
x++  // Postfix increment
--x  // Prefix decrement
x--  // Postfix decrement
```

Assignment Operators

```
int x = 5;
x += 5;  // Equivalent to x = x + 5
x -= 3;  // Equivalent to x = x - 3
x *= 2;  // Equivalent to x = x * 2
x /= 4;  // Equivalent to x = x / 4
x %= 3;  // Equivalent to x = x % 3
```

Bitwise Operators

```
&   // AND
|   // OR
```

```
^   // XOR
~   // NOT _ One's complement
<<  // Left shift
>>  // Right shift
```

Relational Operators

```
<   >   <=  >=  !=  ==
// Note: true = any non-zero value (1, 5, 90...)
//       false = 0
```

Logical Operators

```
&&  // Logical AND
||  // Logical OR
!   // Logical NOT
```

Control Structures

Switch Statement

```
switch(variable) {
    case 1:
        printf("Option 1");
        break;
    case 2:
        printf("Option 2");
        break;
    default:
        printf("Default option");
        break;
}
```

Example Programs

Basic Input/Output

```
#include <stdio.h>

int main() {
```

```
    printf("Name:\t\"Fares Hesham\"\n");
    printf("Faculty:\t'Electronics and Communication Engineering'\n");
    printf("University:\t\"Zagazig\"\n");

    int length, width;
    printf("Enter length: ");
    scanf("%d", &length);
    printf("Enter width: ");
    scanf("%d", &width);
    printf("Area: %d\n", length * width);

    return 0;
}
```

Prime Number Check

```
#include <stdio.h>

int main() {
    int n = 19;
    int count = 0;

    if (n <= 1) {
        printf("%d is NOT prime\n", n);
    } else {
        // Count divisors
        for (int i = 1; i <= n; i++) {
            if (n % i == 0) {
                count++;
            }
        }

        if (count > 2) {
            printf("%d is NOT prime\n", n);
        } else {
            printf("%d is prime\n", n);
        }
    }

    return 0;
}
```

Another IF statement

```c
#include <stdio.h>

int main(){
    int var = 75;
    int var2 = 56;
    int num;
    num = sizeof(var) ? (var2 > 23 ? ((var ==75) ? 'a' :0) :0) :0;
    printf("%d",num);
}
```

Switch Statement example

```c
#include <stdio.h>

int main(){
    int id;
    printf("enter id : ");
    scanf("%d", &id);  // FIXED: use &id
    switch (id){
        case 1:
            printf("Wellcome Fares");
            break;
        case 2:
            printf("Wellcome Ahmed");
            break;
        case 3:
            printf("Wellcome Mohammed");
            break;
        default:
            printf("Invalid id");
            break;
}
```

Escape Sequences

Sequence	Description
`\t`	Tab (4 spaces)
`\n`	New line
`\'`	Single quote

Sequence	Description
`\"`	Double quote
`\v`	Vertical tab
`\\`	Backslash

Engineering Challenges

Modern embedded systems must balance:

- Size: Minimize physical footprint
- Cost: Optimize for production economics
- Configurability: Flexible functionality
- Power Consumption: Energy efficiency

Key Concepts

- Tool Chain: Software that converts source files (.c, .h) into executable files (.exe)
- Compiler: Translates high-level C code into machine code
- MISRA Rules: Coding standards for safety-critical systems

Next Steps

The foundation established here will support:

- Advanced C programming concepts
- Microcontroller programming
- Real-time embedded systems development
- Hardware-software integration

Session 1 Tasks

Task_1 Star pyramid

```c
#include <stdio.h>

int main(){
    printf("    *\n");
    printf("   ***\n");
```

```
    printf("  *****\n");
    printf(" *******\n");
    printf("*********\n");
// OR could use for loop (i) represent row number ,(space) represent the
number of spaces added in each row , (k) make an odd number of *
   int i, space, k = 0;
   for (i = 1; i <= 5; ++i, k = 0) {
      for (space = 1; space <= 5 - i; ++space) {
         printf(" ");
      }
      while (k != 2 * i - 1) {
         printf("*");
         ++k;
      }
      printf("\n");
   }
   return 0;
```

Task_2 Output print

```
#include <STDIO.H>

int main(){
    int num1 , num2, num3;
    printf("Enter the num1: ");
    scanf("%d", &num1);

    printf("Enter the num2: ");
    scanf("%d", &num2);

    printf("Enter the num3: ");
    scanf("%d", &num3);

    printf("%d\n",num3);
    printf("%d\n",num2);
    printf("%d\n",num1);
}
```

Task_3 Output Arithmetic operation

```
#include <STDIO.H>
```

```c
int main(){
    int num1, num2, sum, diff, prod, div, and, or, xor  ;
    printf("Enter the num1: ");
    scanf("%d", &num1);

    printf("Enter the num2: ");
    scanf("%d", &num2);

    sum = num1 + num2;
    diff = num1 - num2;
    prod = num1 * num2;
    div = num1 / num2;
    and = num1 & num2;
    or = num1 | num2;
    xor = num1 ^ num2;

    printf("num1 + num2 = %d\n",sum);
    printf("num1 - num2 = %d\n",diff);
    printf("num1 * num2 = %d\n",prod);
    printf("num1 / num2 = %d\n",div);
    printf("num1 & num2 = %d\n",and);
    printf("num1 | num2 = %d\n",or);
    printf("num1 ^ num2 = %d\n",xor);
}
```

Task_4 Another IF statement

```
A ? B : c;

Evaluates its first operand, and, if the resulting value is not equal to
zero, evaluates its second operand. Otherwise,

it evaluates its third operand, as shown in the following example:

    a = b ? c : d;
    condition ? value_if_true : value_if_false

is equivalent to:

    if (b)
    a = c;
    else
    a = d;
```

```
    ex1:
    int x = 5;
    int y = 42;
    printf("%i, %i\n", 1 ? x : y, 0 ? x : y);       // Outputs "5, 42"


    ex2: The conditional operator can be nested. For example, the
following code determines the biggest of three numbers:
    int a = 8;
    int b = 6;
    int c = 4;
    big= a > b ? (a > c ? a : c): (b > c ? b : c);
         true    (true)                              //out a{8}
```

Task_5 ID based Welcome

```c
#include <stdio.h>

int main(){
    int id;
    printf("enter id : ");
    scanf("%d", &id);  // FIXED: use &id

    switch (id){
        case 1201:
            printf("Wellcome Fares");
            break;
        case 1202:
            printf("Wellcome Ahmed");
            break;
        case 1203:
            printf("Wellcome Mohammed");
            break;
        default:
            printf("Error : Invalid id.");
            break;
    }
}
```

Task_6 input semi array check

```c
#include <STDIO.H>

int main(){
    int num1, num2, num3, num4, num5, num6, num7, num8, num9, num10,
check ;
    printf("Enter the 10 numbers consecutively : ");
    scanf("\n%d\n%d\n%d\n%d\n%d\n%d\n%d\n%d\n%d\n%d", &num1, &num2,
&num3, &num4, &num5, &num6, &num7, &num8, &num9, &num10);

    printf("Enter the number you need to check : ");
    scanf("%d", &check);

    if (check == num1)
        printf("the check number is one of the 10 numbers");
    else if (check == num2)
        printf("the check number is one of the 10 numbers");
    else if (check == num3)
        printf("the check number is one of the 10 numbers");
    else if (check == num4)
        printf("the check number is one of the 10 numbers");
    else if (check == num5)
        printf("the check number is one of the 10 numbers");
    else if (check == num6)
        printf("the check number is one of the 10 numbers");
    else if (check == num7)
        printf("the check number is one of the 10 numbers");
    else if (check == num8)
        printf("the check number is one of the 10 numbers");
    else if (check == num9)
        printf("the check number is one of the 10 numbers");
    else if (check == num10)
        printf("the check number is one of the 10 numbers");
    else
        printf("Check number is Not one of the 10 numers");
}
```

Task_7 Mark grading system

```c
#include <STDIO.H>

int main(){
    int grade;
    printf("Enter the grade: ");
```

```
    scanf("%d", &grade);

    if (grade >= 85 && grade <= 100)
        printf("your grade is : excellent");
    else if (grade <= 85 && grade >= 75)
        printf("your grade is : Very good");
    else if (grade <= 75 && grade >= 65)
        printf("your grade is : Good");
    else if (grade <= 65 && grade >= 50)
        printf("your grade is : normal");
    else if (grade < 50 && grade >= 0)
        printf("your grade is : Faild");
    else
        printf("your grade is : Invalid");
}
```