

Module 2 – Introduction to Programming

1. Research and provide three real-world applications where C programming is extensively used, such as in embedded systems, operating systems, or game development.

- Here are **three real-world applications** where **C programming** is extensively used:

1. Operating Systems (OS Development)

- **Use of C:** Most operating systems, including **UNIX, Linux, and Windows**, are written in C or have large portions written in C.
- **Why C?**
 - Provides low-level access to memory.
 - Offers high performance and efficiency.
 - Portable across different hardware architectures.
- **Example:** The **Linux kernel** is written almost entirely in C, and even parts of Windows are implemented in C.

2. Embedded Systems

- **Use:** C is widely used to program **microcontrollers and embedded devices** that control appliances, automobiles, and industrial machines.
- **Reason:** C provides low-level hardware control with high efficiency, which is crucial for resource-constrained systems.
- **Examples:**
 - **Automobile control systems** (airbag sensors, anti-lock braking systems).
 - **Consumer electronics** like washing machines, microwave ovens, and smartwatches.
 - **Medical devices** such as pacemakers and diagnostic equipment.

3. Game Development (Game Engines & Graphics)

- **Use of C:** Early and modern games still rely on C for performance-critical sections. Game engines are often built using C and C++.
 - **Why C?**
 - High execution speed (crucial for rendering graphics and physics).
 - Ability to optimize system resources.
 - **Example:**
 - **Doom, Quake**, and other classic games were written in C.
 - Many modern game engines like **Unreal Engine** have C at their core.
-

2. Install a C compiler on your system and configure the IDE. Write your first program to print "Hello, World!" and run it.

Step 1: Install a C Compiler

A compiler is required to convert C code into machine language.

Windows:

- Download **MinGW** (Minimalist GNU for Windows) → MinGW-w64
- Install and add the bin folder path (e.g., C:\MinGW\bin) to your **System Environment Variables** → **PATH**.

(This installs gcc, g++, and make tools).

Step 2: Install an IDE (Integrated Development Environment)

You can use:

- **Code::Blocks** (lightweight, simple for beginners).
- **DevC++** (classic, easy for C).
- **VS Code** (modern, needs C/C++ extension).

Step 3: Write Your First C Program

Open your IDE → create a new C project/file → name it HELLO.c → type this code:

```
Assignment > C Hello.c > main()
1  //TO PRINT HELLO WORLD
2  #include<stdio.h>
3  void main()
4  {
5      printf("HELLO WORLD");
6  }
```

Step 4: Compile and Run

```
PS C:\4Aug_Python_Shifa> gcc .\Assignment\Hello.c
PS C:\4Aug_Python_Shifa> .\a.exe
HELLO WORLD
PS C:\4Aug_Python_Shifa> █
```

3. Write a C program that includes variables, constants, and comments. Declare and use different data types (int, char, float) and display their values.

```
Assignment > C lab_3.c > main()
1 // Write a C program that includes variables, constants, and comments.
2 //Declare and use different data types (int, char, float) and display their values.
3 #include<stdio.h>
4 void main()
5 {
6     int radius=2;
7     const float pie=3.14;
8     char starting_name='S';
9     float area;
10    printf("My name starts with %c",starting_name);
11    area=pie*radius*radius;
12    printf("\nRadius is %d",radius);
13    printf("\nArea of circle is %f",area);
14 }
```

4. Write a C program that accepts two integers from the user and performs arithmetic, relational, and logical operations on them. Display the results.

Assignment > C lab_4.c > main()

```
1 // Write a C program that accepts two integers from the user and performs
2 // arithmetic, relational, and logical operations on them. Display the results
3 #include <stdio.h>
4 void main()
5 {
6     int a, b, result; // VARIABLE DECLARATION
7     printf("Enter the value of a::");
8     scanf("%d", &a);
9     printf("Enter the value of b::");
10    scanf("%d", &b);
11    // ARITHMETIC OPERATORS
12    printf("\nADDITION IS %d", a + b);
13    printf("\nSUBTRACTION IS %d", a - b);
14    printf("\nMULTIPLICATION IS %d", a * b);
15    printf("\nDIVISION IS %d", a / b);
16    printf("\nMODULO IS %d", a % b);
17    // RELATIONAL OPERATOR
18    printf("\n");
19    printf("a>b :%d\n", a > b);
20    printf("a<b :%d\n", a < b);
```

```
21    printf("a==b :%d\n", a == b);
22    // LOGICAL OPERATOR
23    int c = 20;
24    int ans;
25    ans = a > b && a > c;
26    printf("Ans of AND operator is %d", ans);
27    ans = a > b || a < c;
28    printf("\nAns of OR operator is %d", ans);
29    ans = !(a > b);
30    printf("\nAns of NOT operator is %d", ans);
31 }
```

5. Write a C program to check if a number is even or odd using an if-else statement. Extend the program using a switch statement to display the month name based on the user's input (1 for January, 2 for February, etc.).

Assignment > C lab_5.c > main()

```
1 // o Write a C program to check if a number is even or odd using an if-else
2 // statement. Extend the program using a switch statement to display the month
3 // name based on the user's input (1 for January, 2 for February, etc.).
4 #include <stdio.h>
5 void main()
6 {
7     int num;
8     printf("Enter the number::");
9     scanf("%d", &num);
10    if (num % 2 == 0)
11    {
12        printf("Number is even");
13    }
14    else
15    {
16        printf("Number is odd");
17    }
18    int month;
19    printf("\nENTER MONTH NUMBER::");
20    scanf("%d", &month);
```

```
21    switch (month)
22    {
23        case 1:
24            printf("JANUARY");
25            break;
26
27        case 2:
28            printf("FEBRUARY");
29            break;
30
31        case 3:
32            printf("MARCH");
33            break;
34
35        case 4:
36            printf("APRIL");
```

```
39     case 5:
40         printf("MAY");
41         break;
42
43     case 6:
44         printf("JUNE");
45         break;
46
47     case 7:
48         printf("JULY");
49         break;
50
51     case 8:
52         printf("AUGUST");
53         break;
54
55     case 9:
56         printf("SPETEMBER");
```

```
54
55     case 9:
56         printf("SPETEMBER");
57         break;
58
59     case 10:
60         printf("OCTOBER");
61         break;
62
63     case 11:
64         printf("NOVEMBER");
65         break;
66
67     case 12:
68         printf("DECEMBER");
69         break;
70
71     default:
72         printf("WRONG NUMBER");
73 }
74 }
```

6. Write a C program to print numbers from 1 to 10 using all three types of loops (while, for, do-while).

Assignment > C lab_6.c > main()

```
1 // Write a C program to print numbers from 1 to 10 using all three types of loops.
2 #include <stdio.h>
3 void main()
4 {
5     int i;
6     printf("USING FOR LOOP\n");
7     for (i = 1; i <= 10; i++)
8     {
9         printf("%d", i);
10        printf("\n");
11    }
12    printf("USING WHILE LOOP\n");
13    i = 1;
14    while (i <= 10)
15    {
16        printf("%d", i);
17        printf("\n");
18        i++;
19    }
```

```
20        printf("USING DO WHILE LOOP\n");
21        int j = 1;
22        do
23        {
24            printf("%d", j);
25            printf("\n");
26            j++;
27        } while (j <= 10);
28    }
```

7. Write a C program that uses the break statement to stop printing numbers when it reaches 5. Modify the program to skip printing the number 3 using the continue statement.

Assignment > C lab_7.c > main()

```
1  /*Write a C program that uses the break statement to stop printing numbers
2  when it reaches 5. Modify the program to skip printing the number 3 using the
3  continue statement.*/
4  #include<stdio.h>
5  void main()
6  {
7      int i;
8      printf("USING BREAK STATEMENT\n");
9      for(i=1;i<=7;i++)
10     {
11         if(i==5)
12         {
13             break;    //BREAK ENCOUNTERS WHEN IT REACHES 5..
14         }
15         printf("%d ",i);
16     }
17     printf("\nUSING CONTINUE STATEMENT\n");
18
19     int j;
20     for(j=1;j<=7;j++)
21     {
22         if(j==3)
23         {
24             continue;
25         }
26         printf("%d ",j);
27     }
28 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\4Aug_Python_Shifa> gcc .\Assignment\lab_7.c

PS C:\4Aug_Python_Shifa> .\a.exe

USING BREAK STATEMENT

1 2 3 4

USING CONTINUE STATEMENT

1 2 4 5 6 7

PS C:\4Aug_Python_Shifa> █

8. Write a C program that calculates the factorial of a number using a function. Include function declaration, definition, and call.

```
Assignment > C lab_8.c > fact(int)
1 // 8. Write a C program that calculates the factorial of a number using a function.
2 // Include function declaration, definition, and call.
3 #include <stdio.h>
4 int fact(int); // FUNCTION DECLARATION
5 void main()
6 {
7     int result;
8     int num;
9     printf("Enter the value of num::");
10    scanf("%d", &num);
11    result = fact(num); //FUNCTION CALL
12    printf("Factorial is %d", result);
13 }
14 int fact(int a) //FUNCTION DEFINITION
15 {
16     int fact = 1;
17     int i = 1;
18     while (i <= a)
19     {
20         fact = fact * i;
21         i++;
22     }
23     return fact;
24 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\4Aug_Python_Shifa> gcc .\Assignment\lab_8.c
PS C:\4Aug_Python_Shifa> .\a.exe
Enter the value of num::5
Factorial is 120
PS C:\4Aug_Python_Shifa> █
```

9. Write a C program that stores 5 integers in a one-dimensional array and prints them. Extend this to handle a two-dimensional array (3x3 matrix) and calculate the sum of all elements.

Assignment > C lab_9.c > main()

```
1  // Write a C program that stores 5 integers in a one-dimensional array and prints
2  // them. Extend this to handle a two-dimensional array (3x3 matrix) and
3  // calculate the sum of all elements
4  #include <stdio.h>
5  void main()
6  {
7      int a[5];
8      int i;
9      for (i = 0; i < 5; i++)
10     {
11         printf("ENTER ELEMENTS::");
12         scanf("%d", &a[i]); // STORES 5 ELEMENT
13     }
14
15     for (i = 0; i < 5; i++)
16     {
17         printf("%d ", a[i]); // PRINTS 5 ELEMENT
18     }
```

```
19     int b[3][3];
20     int j;
21     int sum = 0;
22     printf("\n3*3 MATRIX ELEMENT>>>>\n");
23     for (i = 0; i < 3; i++)
24     {
25         for (j = 0; j < 3; j++)
26         {
27             scanf("%d", &b[i][j]);
28         }
29     }
30     for (i = 0; i < 3; i++)
31     {
32         for (j = 0; j < 3; j++)
33         {
34             printf("%d\t", b[i][j]);
35         }
36         printf("\n");
37     }
38     for (i = 0; i < 3; i++)
39     {
40         for (j = 0; j < 3; j++)
41         {
42             sum = sum + b[i][j];
43         }
44     }
45     printf("SUM OF ALL ELEMENTS OF MATRIX IS %d", sum);
46 }
```

```
PS C:\4Aug_Python_Shifa> gcc .\Assignment\lab_9.c
PS C:\4Aug_Python_Shifa> .\a.exe
ENTER ELEMENTS::1
ENTER ELEMENTS::2
ENTER ELEMENTS::3
ENTER ELEMENTS::4
ENTER ELEMENTS::5
1 2 3 4 5
```

```
3*3 MATRIX ELEMENT>>>>
1
2
3
4
5
6
7
8
9
1      2      3
4      5      6
7      8      9
SUM OF ALL ELEMENTS OF MATRIX IS 45
PS C:\4Aug_Python_Shifa>
```

10. Write a C program to demonstrate pointer usage. Use a pointer to modify the value of a variable and print the result.

```
Assignment > C lab_10.c > main()
1 //Write a C program to demonstrate pointer usage.
2 // Use a pointer to modify the value of a variable and print the result.
3 #include<stdio.h>
4 void main()
5 {
6     int *ptr;
7     int age=19;
8     ptr=&age; //STORES AGE ADDRESS
9     printf("YOUR AGE IS %d",age);
10    printf("\nYOUR AGE VALUE'S ADDRESS IS %p",ptr);
11    printf("\nYOUR POINTER VALUE IS %d",*ptr); //PRINTS POINTER VALUE
12 }
```

```
PS C:\4Aug_Python_Shifa> gcc .\Assignment\lab_10.c
PS C:\4Aug_Python_Shifa> .\a.exe
YOUR AGE IS 19
YOUR AGE VALUE'S ADDRESS IS 0061FF18
YOUR POINTER VALUE IS 19
PS C:\4Aug_Python_Shifa> █
```

11. Write a C program that takes two strings from the user and concatenates them using `strcat()`. Display the concatenated string and its length using `strlen()`.

```
Assignment > C lab_11.c > main()
```

```
1 // Write a C program that takes two strings from the user and concatenates them
2 // using strcat(). Display the concatenated string and its length using strlen().
3 #include <stdio.h>
4 #include <string.h>
5 void main()
6 {
7     char s[20];
8     char b[20];
9     printf("ENTER STRING1::");
10    scanf("%s", &s);
11    printf("ENTER STRING1::");
12    scanf("%s", &b);
13    strcat(s, b);
14    printf("%s ", s);
15    int len = strlen(s);
16    printf("\n");
17    printf("%d", len);
18 }
```



```

26     printf("<<<<<<<<<<STUDENTS  DETAILS>>>>>>>>>");
27     for (i = 0; i < 3; i++)
28     {
29         printf("\nSTUDENT ROLL NO IS %d", s[i].roll_no);
30         printf("\nSTUDENT NAME IS %s",s[i].name);
31         printf("\nSTUDENT MARKS IS %d",s[i].marks);
32         printf("\n");
33     }
34 }

```

```

PS C:\4Aug_Python_Shifa> gcc .\Assignment\lab_12.c
PS C:\4Aug_Python_Shifa> .\a.exe
<<<<<<<<<<<ENTER STUDENTS DETAILS>>>>>>>>>
ENTER STUDENT ROLL NO::101
ENTER STUDENT NAME::SHIFA
ENTER STUDENT MARKS::70

ENTER STUDENT ROLL NO::102
ENTER STUDENT NAME::BAGWAN
ENTER STUDENT MARKS::80

ENTER STUDENT ROLL NO::103
ENTER STUDENT NAME::ABCD
ENTER STUDENT MARKS::70
<<<<<<<<<<<STUDENTS  DETAILS>>>>>>>>>
STUDENT ROLL NO IS 101
STUDENT NAME IS SHIFA
STUDENT MARKS IS 70

STUDENT ROLL NO IS 102
STUDENT NAME IS BAGWAN
STUDENT MARKS IS 80

STUDENT ROLL NO IS 103
STUDENT NAME IS ABCD
STUDENT MARKS IS 70
PS C:\4Aug_Python_Shifa>

```

13. Write a C program to create a file, write a string into it, close the file, then open the file again to read and display its contents.

Assignment > C lab_13.c > main()

```
1 // Write a C program to create a file, write a string into it, close the file, then
2 // open the file again to read and display its contents.
3 #include <stdio.h>
4 #include <string.h>
5 void main()
6 {
7     FILE *fptr;
8     fptr = fopen("HELLO.txt", "w");
9     if (fptr == NULL)
10     {
11         printf("ERROR OPENING IN FILE");
12     }
13     else
14     {
15         fputs("HELLO MYSELF SHIFA BAGWAN", fptr);
16         fclose(fptr);
17     }
18
19     fptr = fopen("HELLO.txt", "r");
20     char ch;
```

```
19     fptr = fopen("HELLO.txt", "r");
20     char ch;
21     // char ch = fgetc(fptr);
22     if (fptr == NULL)
23     {
24         printf("ERROR OPENING IN FILE");
25     }
26     else
27     {
28         while ((ch = fgetc(fptr)) != EOF)
29         {
30             printf("%c", ch);
31         }
32     }
33     fclose(fptr);
34 }
```

```
PS C:\4Aug_Python_Shifa> gcc .\Assignment\lab_13.c
PS C:\4Aug_Python_Shifa> .\a.exe
HELLO MYSELF SHIFA BAGWAN
PS C:\4Aug_Python_Shifa> █
```
