Assignment -01

Experiment 1: Installation, Environment Setup and starting with C language

Aim:

To learn how to:

- Install a C compiler
- Set up the environment
- Write and run basic C programs

How to Set Up C Programming:

It depends on whether you are using Windows, Linux, or macOS.

1. On Windows:

Option A: Using Code::Blocks

1. Download **Code::Blocks with MinGW** from codeblocks.org.
2. Install it → it already includes the **GCC compiler**.
3. Open Code::Blocks → Create a new project → Select **C Console Application**.
4. Write code → Press **F9** to compile and run.

Option B: Using VS Code

1. Install **Visual Studio Code** from code.visualstudio.com.
2. Install **MinGW (GCC)**:
    - Download MinGW from MinGW-w64.
    - Install and add its bin folder path to **System Environment Variables → PATH**.
3. Open VS Code → Install **C/C++ extension** by Microsoft.

4. Write a .c file → Open terminal → Run:

```
gcc program.c -o program.exe
./program.exe
```

2. On Linux (Ubuntu, Debian, Fedora, etc.)

1. Open Terminal.
2. Install GCC:

```
sudo apt update
sudo apt install build-essential   # for Ubuntu/Debian
```

or

```
sudo dnf groupinstall "Development Tools"   # for Fedora
```

```
nano hello.c
```

3. Write code in a file:

Example:

4. Compile and run:

```
#include <stdio.h>
int main() {
    printf("Hello, C!\n");
    return 0;
}
```

```
gcc hello.c -o hello
./hello
```

---

3. On macOS:

1. Open **Terminal**.
2. Install Xcode Command Line Tools:

```
xcode-select --install
```

(This includes GCC/Clang.)

3. Write a program:

```
nano hello.c
```

```
gcc hello.c -o hello
./hello
```

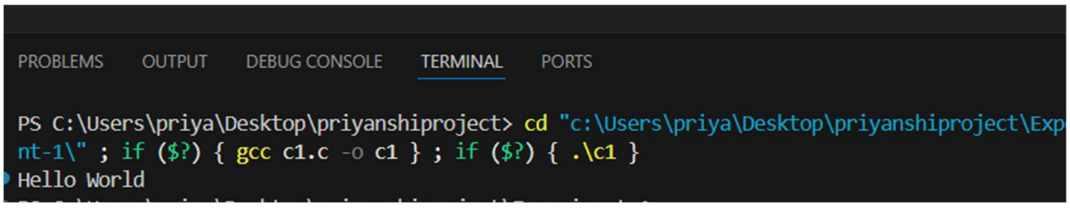
4. Compile and run:

Now you can write, compile, and run **C programs** on your system.

Q1. Write a C program to print “Hello World”.

Code:

Output:



Summary:

- The `#include <stdio.h>` is a preprocessor command that comes as the first statement in our code.
- The `#include` statement tells the compiler to include the standard input/ output library or header file(stdio.h) in the program.
- `main()` is the entry point of the program.
- `printf("Hello World");` prints text on the screen.
- `return 0;` indicates successful execution.

Q2. Write a C Program to print the address in multiple lines (newline).

Code:

```c
//Write a C Program to print the address in multiple lines (newline).


#include <stdio.h>
int main()
{
    //prints the address in multiple line
    printf("Sachin\n");
    printf("Dhanapur, chanduali\n");
    printf("Uttar Pradesh\n");
    printf("India\n");

    return 0;
}
```

Output:

```
 .\c2 }
Sachin
Dhanapur, chanduali
Uttar Pradesh
India
```

Summary:

- \n is used for printing on the next line.
- Each printf prints one line of the address.

Q3. Write a program that prompts the user to enter their name and age.

Code:

```c
//Write a program that prompts the user to enter their name and age.


#include <stdio.h>
int main()
{
    char name[30];
    int age;

    //input the name and age
    printf("Enter the name: ");
    scanf("%s", name);

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

    //prints the result
    printf("Hello %s, your age is %d.", name, age);

    return 0;
}
```

Output:

```
Enter the name: sachin
Enter the age: 20
Hello sachin, your age is 20.
```

Summary:

- scanf("%s", name); takes string input (stops at space).
- scanf("%d", &age); takes integer input.
- %s is format specifier for string, %d for integer.
- We use & for variables except for arrays (like name).
- Name with spaces (e.g., *Sachin*) won't work properly.
- Does not validate negative ages or non-numeric input.

Q4. Write a C program to add two numbers, take numbers from user.

Code:

```c
//Write a C program to add two numbers, take numbers from user.


#include <stdio.h>
int main()
{
    int num1, num2, sum;

    //input two numbers
    printf("Enter the number: ");
    scanf("%d %d", &num1, &num2);

    //calculates the result
    sum = num1 + num2;

    //prints the output
    printf("sum = %d", sum);

    return 0;
}
```

Output:

```
Enter the number: 28 10
sum = 38
```

Summary:

- The program takes two integers as input.
- `sum = num1 + num2;` performs addition.
- Input and output are handled by `scanf()` and `printf()`.