

PROGRAMMING FUNDAMENTALS

(ASSIGNMENT # 03)

(29 NOV 2025)

BY

ZAINAB MALIK

ROLL# 25017119-012

CS-102(PROGRAMMING FUNDAMENTALS) BS CS 5th

SEMESTER INDUCTION SECTION A

SUBMITTED TO

MR.

TASADDAQ LATIF

DEPARTMENT OF COMPUTER SCIENCE



QUESTION #01:

Find the errors in the following codes, if no error then write output.

(a)

```
#include<iostream>
using namespace std;
int main(){
    int I, j;
    for(I=1; I>7; I++)
        cout<<i;
    cout<<"End of Program";
    return 7;
}
```

SOLUTION:

- **Error:** The loop condition $I > 7$ is incorrect. It should be $I < 7$ because I starts at 1 and needs to be less than 7 to print numbers 1 to 6.
- **Output if corrected:** 123456 End of program.

(b)

```
#include<iostream>
using namespace std;
int main(){
    int num1=5, num2=10;
    if(num1<num2)
        cout << "num1 is smallest";
        num1=num2;
    else
        cout << "num2 is smallest";
    return 0;
}
```

SOLUTION:

- **Error:** Missing { } after $\text{if}(\text{num1} < \text{num2})$
So $\text{num1} = \text{num2}$; executes always, and else become unmatched.

- **Output:** num1 is smallest.

(c)

```
#include<iostream>
using namespace std;
int main(){
    int a=0, b=1;
    a=1-b;
    if(a < b;
    b++ + a;
    cout << a << b;
    return 0;
}
```

SOLUTION:

- **Output:** 1222

Explanation:

- a=b++ ; assigns a=1 (post-increment: b becomes 2).
- First cout prints a and b as 12.
- b=++a ; increments a to 2 and assigns b=2 (pre-increment).
- Second cout prints a and b as 22.
- Thus, output is 1222.

(d)

```
#include<iostream>
using namespace std;
int main(){
    cout<<"Enter values";
    int a,b,res;
    cin>>a; cin>>b;
    a=20; b=10;
    a+=res;
    cout<<"res = "<<res;
    return a;
}
```

SOLUTION:

- **Error:** `a+b =res;` is incorrect. It should be **`res=a + b;`** because in C++, assignment is done with the variable on the left.

QUESTION #02:

Define the following:

A) Header files:

Header files are special files in C++ that contain **pre-written code**, such as function declarations, classes, and libraries. They are included in a program using the **#include** statement.

Example:

`#include <iostream>` → allows us to use `cout` and `cin`.

B) Source Code:

Source code is the **original code written by the programmer** in a programming language such as C++. It is saved in a file (like `.cpp`) and later compiled to run the program.

C) Comments:

Comments are **notes written inside a program** to explain the code.

The compiler **ignores** comments; they are only for humans to understand the code.

Non-executable notes used to explain code.

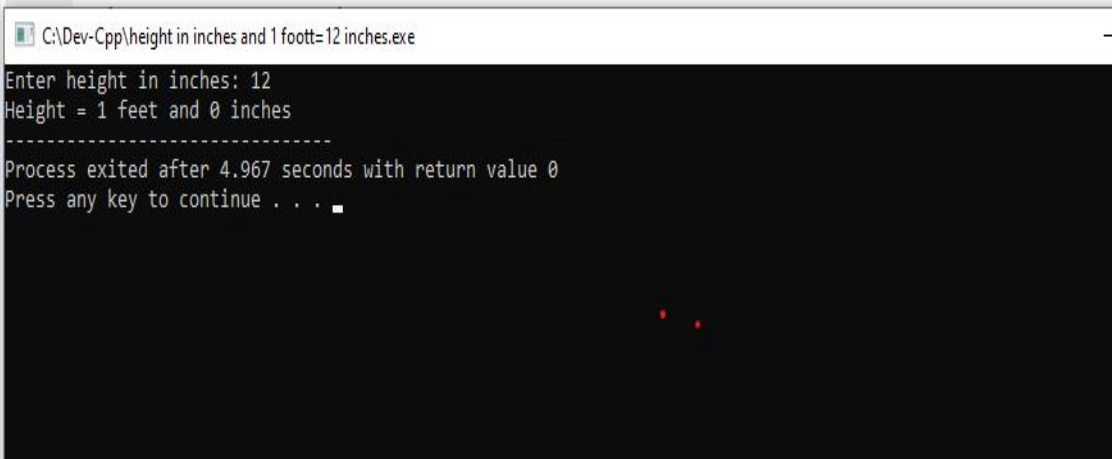
// single line, /* multi-line */.

QUESTION #03:

Write a program that read height in inches. And display the height in feet and inches.

```
#include <iostream>
using namespace std;
int main()
{
    int inches;
    cout << "Enter height in inches: ";
    cin >> inches;
    int feet = inches / 12;
    int remInches = inches % 12;

    cout << "Height = " << feet << " feet and " <<
remInches << " inches";
    return 0;
}
```




QUESTION #04:

Write a program that take two numbers from user. Find if the number is multiple of second number or not. For example if first number is 49 and second number is 7. then it should print that 7 is multiple of 49.

```
#include <iostream>
using namespace std;
int main()
{
    int a, b;
    cout << "Enter two numbers: ";
    cin >> a >> b;

    if (a % b == 0)
        cout << a << " is a multiple of " << b;
    else
        cout << a << " is NOT a multiple of " << b;

    return 0;
}
```



```
enter two numbers:49
7
49is a multiple of7
-----
Process exited after 5.801 seconds with return value 0
Press any key to continue . . .
```

```
enter two numbers:7
49
7is NOT a multiple of49
-----
Process exited after 3.993 seconds with return value 0
Press any key to continue . . . ■
```

QUESTION #05:

Write a program that read 10 numbers from user and display their sum, maximum and minimum number.

```
#include <iostream>
using namespace std;
int main()
{
    int num, sum = 0, max, min;
    cout << "Enter 10 numbers:\n";
    cin >> num;
    sum = max = min = num;

    for(int i = 2; i <= 10; i++)
    {
        cin >> num;
        sum += num;
        if(num > max) max = num;
        if(num < min) min = num;
    }

    cout << "Sum = " << sum << endl;
    cout << "Maximum = " << max << endl;
    cout << "Minimum = " << min << endl;
    return 0;
}
```

```
Enter 10 numbers:
1
2
3
4
5
6
7
8
9
10
Sum = 55
Maximum = 10
Minimum = 1

-----
Process exited after 10.89 seconds with return value 0
Press any key to continue . . .
```

Array Sorting:

Array sorting in C++ means **arranging the elements of an array in a specific order**, usually:

- **Ascending order** (small → large)
 - **Descending order** (large → small)
- makes data easier to search, process, or analyze.

Sorting

Example(Before & After Sorting)

- **Before sorting:**

[7, 2, 9, 1]

- **After sorting (ascending):**

[1, 2, 7, 9]

Types of Sorting in C++

1. Using built-in function

```
sort(arr, arr + n);
```

2. Manual algorithms:

- Bubble sort
- Selection sort
- Insertion sort
- Merge sort
- Quick sort

1. Bubble Sort:

Bubble sort compares adjacent elements and swaps them if they are in the wrong order.

This process repeats until the array becomes sorted.

```
#include <iostream> using namespace std;
```

```
int main() {
```

```
    int arr[5] = {5, 2, 9, 1, 6};
```

```
    for(int i = 0; i < 5; i++){
```

```
        for(int j = 0; j < 4 - i; j++){
```

```
            if(arr[j] > arr[j+1]){
```

```
                int temp = arr[j];
```

```
                arr[j] = arr[j+1];
```

```
                arr[j+1] = temp;
```

```
            }
```

```
        }
```

```
    }
```

```
    for(int i=0; i<5; i++)
```

```
        cout << arr[i] << " ";
```

```
}
```

2. Selection Sort:

Selection sort finds the smallest element from the unsorted part and places it at the beginning of the array.

```
#include <iostream>using namespace std;
int main() {
    int arr[5] = {5, 2, 9, 1, 6};

    for(int i = 0; i < 4; i++){
        int minIndex = i;

        for(int j = i + 1; j < 5; j++){
            if(arr[j] < arr[minIndex])
                minIndex = j;
        }

        int temp = arr[i];
        arr[i] = arr[minIndex];
        arr[minIndex] = temp;
    }

    for(int i=0; i<5; i++)
        cout << arr[i] << " ";
}
```

3. Insertion Sort:

Insertion sort builds a sorted array one element at a time by inserting each element into its correct position.

```
#include <iostream>using namespace std;
int main() {
    int arr[5] = {5, 2, 9, 1, 6};
```

```

for(int i = 1; i < 5; i++){
    int key = arr[i];
    int j = i - 1;

    while(j >= 0 && arr[j] > key){
        arr[j+1] = arr[j];
        j--;
    }
    arr[j+1] = key;
}

for(int i=0; i<5; i++)
    cout << arr[i] << " ";
}

```

4. Built-in sort ()Function:

C++ provides a built-in function `sort()` in `<algorithm>` that can sort arrays in ascending or descending order efficiently.

```

#include <iostream>#include <algorithm>using namespace std;
int main() {
    int arr[5] = {5,2,9,1,6};

    sort(arr, arr + 5);

    for(int i=0; i<5; i++)
        cout << arr[i] << " ";
}

```