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 */.

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)
- Sorting makes data easier to search, process, or analyze.

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] << " ";
}
```