

* Object Oriented Programming (OOP) - Practical Number - 6 (Group - B)

Name:- Kaustubh Shrikant Kabra

Class:- Second Year Engineering

Div:- A

Roll Number:-

Batch:-

Department:- Computer Department

College:- AISSMS's IOIT.

Title:-

Demonstrate function templates for sorting.

Objective:-

- 1> To learn and understand templates.
- 2> To demonstrate function templates for selection sort.

Problem Statement:-

Write a function template selection Sort. Write a program that inputs, sort and output an integer array and a float array.

Outcomes:-

- 1> Student will be able to learn and understand working and use of function template.
- 2> Student will be able to demonstrate function template for selection sort.

Hardware Requirement:-

Any CPU with pentium processor or similar, 256 MB RAM or more, 1GB Hard Disk or more.

Software Requirements:-

64 bit Linux/Windows Operating System, G++ compiler.

Theory:-

Templates-

Templates are a feature of the C++ programming language that allows functions and classes to operate with generic types. This allows a function or to work on many different data types without being rewritten for each one. Templates are the foundation of generic programming, which involves writing code in a way that is independent of any particular type.

Function Template: The general form of a template function definition is shown here:

```
template <typename type> ret-type func-name (parameter list)
```

```
{
```

```
    // body of function
```

```
}
```

Selection Sort-

Selection sort is a simple algorithm. This sorting algorithm is an in-place comparison based algorithm in which the list is divided into two parts, sorted part at left end and unsorted part at right end. Initially sorted part is empty and unsorted part is entire list.

This algorithm is not suitable for large data sets as its average and worst case complexity are of $O(n^2)$ where n are number of items.

Step 1 - Set MIN to location 0.

Step 2 - Search the minimum element in the list.

Step 3 - Swap with value at location MIN.

Step 4 - Increment MIN to point to next element.

Step 5 - Repeat until list is sorted.

Algorithm :-

- 1) Start.
- 2) Declare function template for selection sort.
- 3) Template `<class T> T selection (T a[], T n, {})`.
- 4) In main, declare variable for choice.
- 5) Accept choice from user.
- 6) Call function template for int array x and $n = \text{size}$.
- 7) Display sorted array.
- 8) If choice == 2) accept size and element in float.
- 9) Go to step 6.
- 10) Stop.

Test case:-

Number of element:- 7

Enter elements :- 27 9 -27 7 5 19 -19

After pass 1 :- -27 27 9 7 5 19 -19

After pass 2 :- -27 -19 27 9 7 5 19

After pass 3 :- -27 -19 5 27 9 7 19

After pass 4 :- -27 -19 5 7 27 9 19

After pass 5 :- -27 -19 5 7 9 27 19

After pass 6 :- -27 -19 5 7 9 19 27.

After sorting:- -27 -19 5 7 9 19 27.

Conclusion:-

Here we demonstrated use of function template for selection sort.