

Assignment No 8

Name:- Sameer Manoj Bramhecha

Roll No:- 21115

Batch:- E-1

Performance Date:- 12/11/21

Submission Date:- 16/11/2021

Title:-> Demonstrate function template for sorting algorithm.

Problem Statement:-> Write a function template ^{for} selection sort. Write a program that inputs, sorts and outputs an integer array and a float array.

Learning Objectives:-

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

Learning Outcomes:- After completion of this assignment, students will be able to:

- ① Implement the concept of function template.
- ② Implement the algorithm for selection sort.

S/W or H/W requirements:-

- 1] 64-bit open source ~~Linux~~ Windows
- 2] Open source C++ programming tool like G++/Gcc.

Theory:-

Templates are a feature of the C++ programming language that allows functions and classes to operate with generic types. This allows a function or class 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. A template is a blueprint or formula for creating a generic class or a function. The library containers like iterators and algorithms are examples of generic programming and have been developed using template concept.

There is a single definition of each container, such as vector, but we can define many different kinds of vector for example, `vector<int>` or `vector<string>`.

You can use templates to define functions as well as classes, let us see how they work:

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

```
template <typename type> return-type func-name(parameter list)
{
    // body of function
}
```

Here, `type` is a placeholder name for a data type used by the function. This name can be used within the function definition.

A function template behaves like a function except that the template can have arguments of many different types. In other words, a function template represents a family of functions. The format for declaring function templates with type parameters is:


```
template<class identifier>
```

```
return-type func-name (arglist with at least one type as class identifier);
```

OR.

```
template<typename identifier>
```

```
return-type func-name (arglist with at least one type as type-identifier);
```

Both expressions have the same meaning and behave in exactly the same way. The latter form was introduced to avoid confusion, since a type parameter need not be a class (it can also be a basic type such as int or double).

For example, the C++ Standard Library contains the function template `max(x, y)` which returns the larger of `x` and `y`. That function template could be defined like this:

```
template <typename T>
```

```
inline T max (T a, T b) {
```

```
return a > b ? a : b;
```

```
}
```

This single function definition works with many data types. The usage of a ~~tem~~ function template saves space in the source code file in addition to limiting changes to one function description and making the code easier to read.

A template does not produce smaller object code, though, compared to writing separate functions for all the different data types used in a specific program. For example, if a program uses both an int and a double version of the `max()` function template shown above, the compiler will

Algorithm :-

1) Algorithm for function template:-

- 1) Start
- 2) Define template of class T
- 3) Define function for selection sort
- 4) In selection-sort(), initialize an array with 5 element.
- 5) Declare i, j, min as integer variables.
- 6) Declare template temp.
- 7) Accept 5 nos. according to selected data type.
- 8) Using algorithm ②, perform selection sort on the accepted array.
- 9) Display the sorted array.
- 10) Stop.

2) Algorithm for Selection Sort:-

- 1) Start
- 2) Set min to location 0.
- 3) Search the minimum element in the list.
- 4) Swap the minimum value at location min.
- 5) Increment min to point to next element.
- 6) Repeat until list is sorted.
- 7) Stop.

3) Algorithm for main() :-

- 1) Start
- 2) Display menu and accept user's choice.

3.) If user choice is '1' i.e. 'int', call selection sort template for 'int' values.

4.) If user choice is '2' i.e. 'float', call selection sort template for 'float' values.

5.) If user choice is '3', Exit from code.

6.) Repeat until user enters 3 as choice.

7.) ~~End~~. Stop.

Test Cases:-

Test Case No.	Test Case Description	Input	Expected Output	Actual Output	Result
			Sorted Array:		
1.	Int Array	10	5	5	Pass.
		15	10	10	
		5	153	13	
		13	15	15	
		41	41	41	
2.	Float Array	2.1	1.101	1.101	Pass.
		4.6	1.102	1.102	
		1.101	2.1	2.1	
		1.102	4.2.11	2.11	
		2.11	4.6	4.6.	

Conclusion:- we have demonstrated and understood the use of function template for selection sort.