Assignment - 08 (OOPs)

Roll No. 21123 Batch - SE1(F1)

Div. SE1 DOP: DOS:

Title - Demonstration function template for sorting algorithm.

objectives: 1) To learn and understand templates.
2) To demonstrate function template for selection sort.

Problem statement - Write a function template selection sort Write a program that inputs, sort and outputs an integers array and a float array.

Theory:

Templates are a feature of the c++ programming languague that allows function and classes to operate with generic type. 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 example of generic programming and have been developed using template concept.

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

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

Function template: The general form of a template function definition is show here.

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

Here, a 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 otherwords, a function templates represents a family of function. The format for declaring function. The format for declearing function templates with type parameter is:

```
template <class identifiers>
return-type
fun-name (arglist_with_atleast-one-type_as_class_identifier)
OR
template <typename identifiers>
return-type
fun-name (arglist_with_atleast_one type_as_type_identifies)
```

Both expression 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.

for example, the c++ Std library contains the function template max(x,y) which returns the large of x & y. That function template could be defined like this:

```
template <typname T>
inline T max(Ta,Tb)
 {
    return a>b ? a:b;
 }
```

This single function definition works with many data types. The usage of a 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 obj code, though compared to writing seperate 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 create an object code version of max() that operates on int arguments and another object code version that operates on double arguments. The compilers output will be identical to what would have been produced if the sources code had contained two seperate non-template version of max(), one written to handle int and one written to handle double.

Selection sort:

It is a simple sorting algorithm. This sorting algorithm is a 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.

smallest element is selected from the unsorted array and swapped with the leftmost element and that element become part of sorted array. this process continues moving unsorted array boundary by one element to the right.

This algorithm is not suitable for large data sets as it its average and worst case complexity are of $O(n^2)$ where n are no. of items

step1. set min to location 0
step2. search the minimum element in the list
step3. swap with value at location min
step4. Increment min to point to next element
step5. Repeat until list is sorted.

Algorithm.

1. Define a function template with name input()
2. Accept size number of elements from user for sorting in input() // Total of student.
3. Take the each element in the array.
4. Declare the variable min to hold index position of smallest element in array to sort.
5. Define for loop with loop variable "i" from 0 to size incremented by one.
6. Assign first index position to min variable i.e 0
7. Define second for loop, check if element at index position "j" is smaller than element at index position.
8. If the condition is true, assign min as j else continue.
9. In outer for loop, swap element, swap element at index position "i" with element at index position at min.
10. In main() method, declare variable size to accept size of array.
11. Input number of elements to sort and delare an array of given size.
12. use int as data of array.
13. Call input() fn and pass array name and size ~~of~~.
14. Call sort() function ~~and pass~~.
15. Repeated step1 from 12 to 14 with datatype float
16. End.