EXPERIMENT NO - 14

Title: Write a program to implement function template and class template.

Objectives: To write a generic function using function template and generic class using class template.

Key Concepts: template, function template, typename, class, class template

Theory:

Templates in C++

Template is simple and yet very powerful tool in C++. The simple idea is to pass data type as a parameter so that we don't need to write same code for different data types. For example a software company may need sort() for different data types. Rather than writing and maintaining the multiple codes, we can write one sort() and pass data type as a parameter.

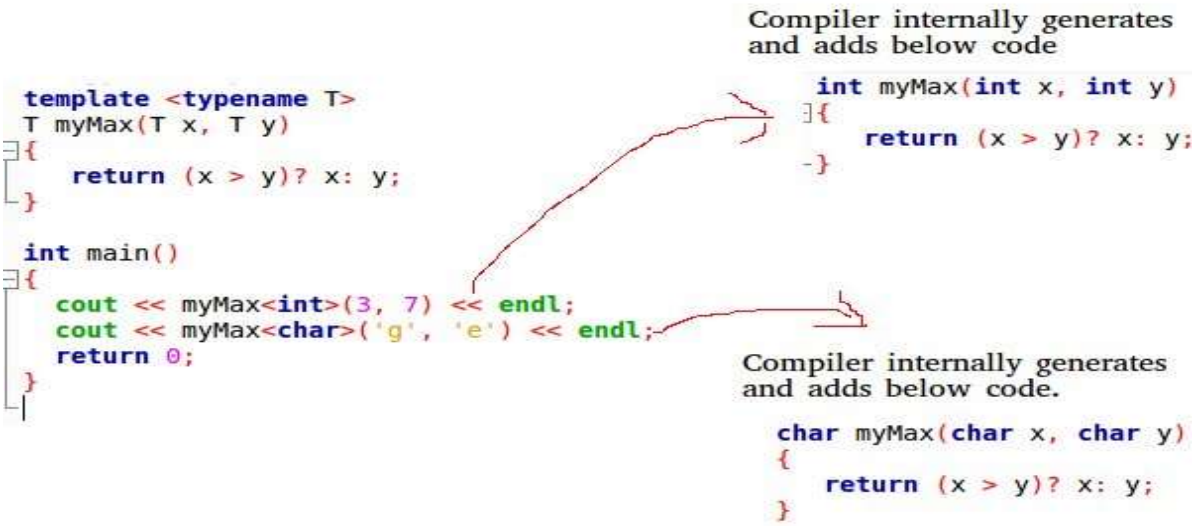
C++ adds two new keywords to support templates: 'template' and 'typename'. The second keyword can always be replaced by keyword 'class'.

How templates work?

Templates are expended at compiler time. This is like macros. The difference is, compiler does type checking before template expansion. The idea is simple, source code contains only function/class, but compiled code may contain multiple copies of same function/class.

Function Templates :

We write a generic function that can be used for different data types.



Class Templates :

Class Templates Like function templates, class templates are useful when a class defines something that is independent of data type. Can be useful for classes like LinkedList, BinaryTre, Stack, Queue, Array, etc.

Example:

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

template <typename T>
class Array {
private:
    T *ptr;
    int size;
public:
    Array(T arr[], int s);
    void print();
};

template <typename T>
Array<T>::Array(T arr[], int s) {
    ptr = new T[s];
    size = s;
    for(int i = 0; i < size; i++)
        ptr[i] = arr[i];
}

template <typename T>
void Array<T>::print() {
    for (int i = 0; i < size; i++)
        cout<<" "<<*(ptr + i);
    cout<<endl;
}

int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    Array<int> a(arr, 5);
    a.print();
    return 0;
}
```

we can pass more than one data types as arguments to templates. we can specify default arguments to templates

A] Problem Statement:

Write a program to implement linear search using function template.

Program Analysis:

- Write a function template for linear search.
- Pass the searching element and input array to the function as parameter.
- Write a searching logic inside function if element found return the flag.
- Test the program for different datatypes.

B] Problem Statement :

Write a program to implement stack using class template.

Program Analysis:

- Create a class template for class
- Perform push and pop operations for stack
- Test the program for any data type.