Chapter 12_Template

Reference : E. Balaguruswamy Object Oriented Programming With C++; chapter-12.

1. What is generic programming? how is it implemented in C++? Or, What is template?
2. What is a generic class and what is its general form?
3. Write a C++ program to manipulate a stack.

Q:1::What is generic programming? how is it implemented in C++? Or, What is template?
Answer:
Generic programming: Generics is the idea to allow type (Integer, String, … etc, and user-defined types) to be a parameter to methods, classes, and interfaces.Entities such as a class or function created using generic programming are called generics.
The method of Generic Programming is implemented to increase the efficiency of the code. Generic Programming enables the programmer to write a general algorithm that will work with all data types. It eliminates the need to create different algorithms if the data type is an integer, string, or character.
Implementing generic programming / Template : Generics can be implemented in C++ using Templates. The template is a 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 the same code for different data types. A template can be used to create a family of functions or classes.So,to implement generic programming template can be used in two ways:

I. Function Template or generic function
II. Class Template or generic class

Function Templates
A function template is a function that contains generic code to operate on different types of data. This enables a programmer to write functions without having to specify the exact type of parameters.Generic functions use the concept of a function template. Generic functions define a set of operations that can be applied to the various types of data.The type of data that the function will operate on depends on the type of data passed as a parameter.
A Generic function is created by using the keyword template. The template defines what the function will do. The syntax for defining a template function is as follows:

```
template<typename t1, typename t2,…>
return-type function-name(t1 arg1,t2 arg2, …)
{
        //Body of function template
}
```
The template keyword tells the compiler that what follows is a template. Here, typename is a keyword and t1,t2,..... are the name of a generic argument. Code link

Q:2::What is a generic class and what is its general form?
Answer:
Using templates programmers can create abstract classes that define the behavior of the class without actually knowing what data type will be handled by the class operations. Such classes are known as class templates. The syntax for creating a class template is as follows:

```
template<typename t1, typename t2,…>
class ClassName
{
        t1 datamember1;
        t2 datamember2;
        …………………….
        //Body of the class
};
```

Syntax for creating an object of the template class is as follows:

```
ClassName<t1,t2,....> ObjectName(params-list);
```

The process of creating an object of a template class or creating a specific class from a class template is called instantiation. The instantiated object of a template class is known as specialization. If a class template is specialized by some but not all parameters it is called partial specialization and if all the parameters are specialized, it is a full specialization.

Code link

Q:3::Write a C++ program to manipulate a stack.
Answer:

Code link