Department of Computer Science & Engineering, NITS
CS212 - Object Oriented Programming Laboratory
Session : January - June 2024

Lab-5: Template

Both object-oriented programming (OOP) and generic programming deal with types that are not known at the time the program is written. The distinction between the two is that OOP deals with types that are not known until run time, whereas in generic programming the types become known during compilation.

Templates are the foundation of generic programming. We can use and have used templates without understanding how they are defined. ***A template is a blueprint or formula for creating classes or functions***.

Assignments:

1. In C++, there is no check on an array index out of bounds. However, during program execution, an array index out of bounds can cause serious problems. Also, in C++, the array index starts at 0.
   Design and implement the class **myArray** that solves the array index out of bounds problem and also allows the user to begin the array index starting at any integer, positive or negative. Every object of type **myArray** is an array of type int . During execution, when accessing an array component, if the index is out of bounds, the program must terminate with an appropriate error message. Consider the following statements:
   myArray< *int* > list(5); // line 1
   myArray< *int* > myList(2, 13); // line 2
   myArray< *int* > yourList(-5, 9); // line 3
   The statement in Line 1 declares list to be an array of 5 components, the component type is int , and the components are: list[0] , list[1] , ... , list[4] ; the statement in Line 2 declares myList to be an array of 11 components, the component type is int , and the components are: myList[2] , myList[3] , ... , myList[12] ; the statement in Line 3 declares yourList to be an array of 14 components, the component type is int , and the components are: yourList[-5] , yourList[-4] , ... , yourList[0] ,... , yourList[8] . Write a program to test the class myArray and redesign the class myArray using class templates so that the class can be used in any application that requires arrays to process data.

2. Create the C++ Function Template named swap so that it has two parameters of the same type. A Template Function created from swap will exchange the values of these two parameters. Write a program to test it.

3. Create the C++ Function Template named multiples so that it has three parameters sum, x, and n. The first two parameters will have the type represented by the function template type parameter WhatKind. n will always be int. The return type is void. All parameters are passed by value except for sum which is passed by reference. A Template Function created from multiples will compute...

   $$sum = 1 + x + 2x + 3x + ... + n$$

   Write a program to test it.

Submission Details :

- Date of Issue : 02-04-2024
- Date of Submission : (Soft Copy, before 2 PM, 09-04-2024)
- Evaluation time : 2 - 5 PM