RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

CSPS103: Object Oriented Programming

Lecture-39

Preeti Singh

Department of Computer Science & Engineering

Rama University, Kanpur

preeti.ru@ramauniversity.ac.in

OBJECTIVES

In this lecture, you will learn to:

- ❖Templates
- ❖Types of Templates
- ❖Function Template
- ❖Example of a function template

TEMPLATES

- A C++ template is a powerful feature added to C++.

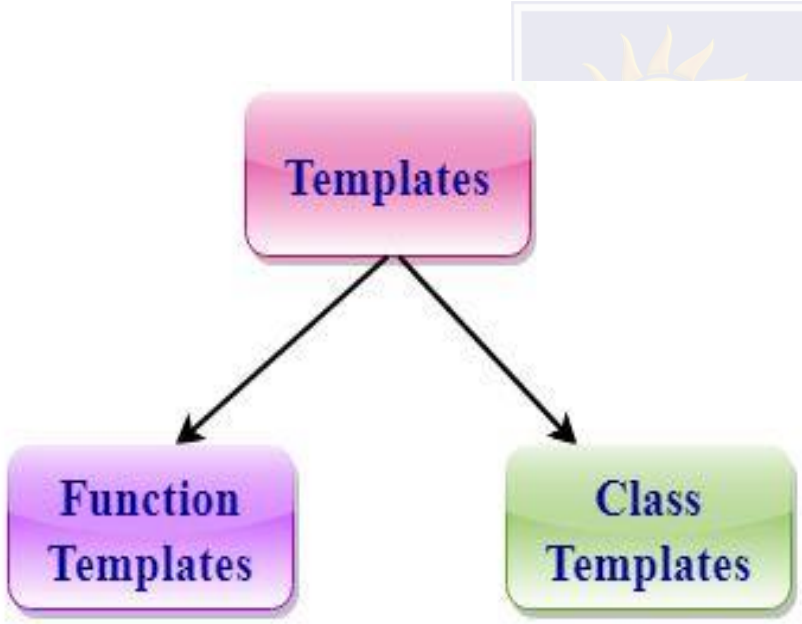
- It allows you to define the generic classes and generic functions and thus provides support for generic programming.

- Generic programming is a technique where generic types are used as parameters in algorithms so that they can work for a variety of data types.

TYPES OF TEMPLATES

Templates can be represented in two ways:

1. Function templates
2. Class templates



FUNCTION TEMPLATE

❑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 the data that the function will operate on depends on the type of the data passed as a parameter.

❑For example, Quick sorting algorithm is implemented using a generic function, it can be implemented to an array of integers or array of floats.

❑A Generic function is created by using the keyword template. The template defines what function will do.

SYNTAX OF FUNCTION TEMPLATE

```
template < class Ttype> ret_type func_name(parameter_list)
{
   // body of function.
}
```

- Where **Ttype:** It is a placeholder name for a data type used by the function.

- **class**: A class keyword is used to specify a generic type in a template declaration.

EXAMPLE OF A FUNCTION TEMPLATE

```
#include <iostream>

template<class T> T add(T &a,T &b)
{
    T result = a+b;
    return result;

}
int main()
{
  int i =2;
  int j =3;
  float m = 2.3;
  float n = 1.2;
  cout<<"Addition of i and j is :"<<add(i,j);
  cout<<'\n';
  cout<<"Addition of m and n is :"<<add(m,n);
  return 0;
}
```

REFERENCES

- Kernighan, Brian W., and Dennis M. Richie. The C Programming Language. Vol. 2. Englewood Cliffs: Prentice-Hall, 1988.
- King, Kim N., and Kim King. C programming: A Modern Approach. Norton, 1996.
- Bjrane Stroustrup, “C++ Programming language”,3rd edition, Pearson education Asia(1997)
- Lafore R.”Object oriented Programming in C++”,4th Ed. Techmedia,New Delhi(2002).
- Yashwant Kenetkar,”Let us C++”,1stEd.,Oxford University Press(2006)
- B.A. Forouzan and R.F. Gilberg,CompilerScience,”A structured approach using C++” Cengage Learning, New Delhi.
- https://www.javatpoint.com/cpp-tutorial
- https://www.tutorialspoint.com/cplusplus/index.htm
- https://ambedkarcollegevasai.com/wp-content/uploads/2019/03/CPP.pdf
- https://onlinecourses.nptel.ac.in/noc20_cs07/unit?unit=3&lesson=19

Multiple Choice Question:

Q1. What is a template?

a) A template is a formula for creating a generic class

b) A template is used to manipulate the class

c) A template is used for creating the attributes

d) A template is used to delete the class

Multiple Choice Question:

Q2. How to declare a template?

a) tem

b) temp

c) template<>

d) temp()

Multiple Choice Question:

Q3. How many types of templates are there in c++?

a) 1

b) 2

c) 3

d) 4

Multiple Choice Question:

Q4. Which are done by compiler for templates?

a) type-safe

b) portability

c) code elimination

d) prototype

Multiple Choice Question:

Q5. What may be the name of the parameter that the template should take?

a) same as template

b) same as class

c) same as function

d) same as member

Summary

In this lecture, you learned that:

- Templates allows you to define the generic classes and generic functions and thus provides support for generic programming
- Generic functions define a set of operations that can be applied to the various types of data.