
RAMA UNIVERSITY

www.ramauniversity.ac.in

# FACULTY OF ENGINEERING & TECHNOLOGY

## CSPS103: Object Oriented Programming

## Lecture-40

Preeti Singh

Department of Computer Science & Engineering

Rama University, Kanpur

preeti.ru@ramauniversity.ac.in

# OBJECTIVES

**In this lecture, you will learn to:**

- ❖Function Templates with Multiple Parameters
- ❖Example of Multiple Parameters
- ❖Overloading a Function Template
- ❖Restrictions of Generic Functions
- ❖Class template
- ❖Example of Class template

# FUNCTION TEMPLATES WITH MULTIPLE PARAMETERS

We can use more than one generic type in the template function by using the comma to separate the list.

## Syntax

```
 template<class T1, class T2,.....>

return_type function_name (arguments of type T1, T2....)

{

    // body of function.

}
```

# EXAMPLE OF MULTIPLE PARAMETERS

```
#include <iostream>


template<class X,class Y> void fun(X a,Y b)
{
    std::cout << "Value of a is : " <<a<< std::endl;
    std::cout << "Value of b is : " <<b<< std::endl;
}


int main()
{
   fun(15,12.3);

   return 0;
}
```

# OVERLOADING A FUNCTION TEMPLATE

We can overload the generic function means that the overloaded template functions can differ in the parameter list.

## Example

```
#include <iostream>


template<class X> void fun(X a)
{
    std::cout << "Value of a is : " <<a<< std::endl;
}
template<class X,class Y> void fun(X b ,Y c)
{
    std::cout << "Value of b is : " <<b<< std::endl;
    std::cout << "Value of c is : " <<c<< std::endl;
}
int main()
{
   fun(10);
   fun(20,30.5);
   return 0;
}
```

# RESTRICTIONS OF GENERIC FUNCTIONS

Generic functions perform the same operation for all the versions of a function except the data type differs.

## Example

```
#include <iostream>
void fun(double a)
{
    cout<<"value of a is : "<<a<<'\n';
}

void fun(int b)
{
    if(b%2==0)
    {
        cout<<"Number is even";
    }
    else
    {
        cout<<"Number is odd";
    }

}

int main()
{
   fun(4.6);
   fun(6);
   return 0;
}
```

# CLASS TEMPLATE

❑Class Template can also be defined similarly to the Function Template.

❑When a class uses the concept of Template, then the class is known as generic class.

## Syntax

```
template<class Ttype>
class class_name
{
  .
  .
}
```

**Ttype** is a placeholder name which will be determined when the class is instantiated

# CLASS TEMPLATE (Contd.)

we create an instance of a class

**class_name<type> ob**;

**where class_name**: It is the name of the class.

**type**: It is the type of the data that the class is operating on.

**ob**: It is the name of the object.

# EXAMPLE OF CLASS TEMPLATE

```
#include <iostream>

template<class T>
class A
{
    public:
    T num1 = 5;
    T num2 = 6;
    void add()
    {
        std::cout << "Addition of num1 and num2 : " << num1+num2<<std::endl;
    }

};

int main()
{
    A<int> d;
    d.add();
    return 0;
}
```

# CLASS TEMPLATE WITH MULTIPLE PARAMETERS

We can use more than one generic data type in a class template, and each generic data type is separated by the comma.

## Syntax

```
template<class T1, class T2, .....>
class class_name
{
   // Body of the class.
}
```

# EXAMPLE OF CLASS TEMPLATE CONTAINS TWO GENERIC DATA TYPES

```
#include <iostream>
     using namespace std;
     template<class T1, class T2>
    class A
    {
         T1 a;
         T2 b;
         public:
        A(T1 x,T2 y)
       {
           a = x;
           b = y;
        }
           void display()
          {
                std::cout << "Values of a and b are : " << a<<" ,"<<b<<std::endl;
           }
      };

      int main()
     {
          A<int,float> d(5,6.5);
          d.display();
          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. How many parameters are legal for non-type template?**

a) 1

b) 2

c) 3

d) 4

## Multiple Choice Question:

**Q2. What is a function template?**

a) creating a function without having to specify the exact type

b) creating a function with having an exact type

c) creating a function without having blank spaces

d) creating a function without class

## Multiple Choice Question:

**Q3. Which is used to describe the function using placeholder types?**

a) template parameters

b) template type parameters

c) template type

d) type parameters

## Multiple Choice Question:

**Q4. Pick out the correct statement.**

a) you only need to write one function, and it will work with many different types

b) it will take a long time to execute

c) duplicate code is increased

d) it will take a long time to execute & duplicate code is increased

## Multiple Choice Question:

**Q5. What is the syntax of class template?**

a) template <paramaters> class declaration

b) Template <paramaters> class declaration

c) temp <paramaters> class declaration

d) Temp <paramaters> class declaration

# Summary

## In this lecture, you learned that:

- C++ supports a powerful feature known as a template to implement the concept of generic programming.
- A template allows us to create a family of classes or family of functions to handle different data types.
- Template classes and functions eliminate the code duplication of different data types and thus makes the development easier and faster.
- Multiple parameters can be used in both class and function template.
- Template functions can also be overloaded.