RAMA UNIVERSITY

www.ramauniversity.ac.in

# FACULTY OF ENGINEERING & TECHNOLOGY

## CSPS103: Object Oriented Programming

## Lecture-24

Preeti Singh
Department of Computer Science & Engineering
Rama University, Kanpur

preeti.ru@ramauniversity.ac.in

# OBJECTIVES

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

- ❖ **Function Overloading**
- ❖ **Example Function Overloading**

# FUNCTION OVERLOADING

- Function Overloading is defined as the process of having two or more function with the same name, but different in parameters is known as function overloading in C++.

- In function overloading, the function is redefined by using either different types of arguments or a different number of arguments.

- It is only through these differences compiler can differentiate between the functions.

# EXAMPLE : FUNCTION OVERLOADING

```
#include<iostream.h>
#include<conio.h>
int sum(int p,int q,int r);
double sum(int l,double m);
float sum(float p,float q)
int main(){
cout<<"sum="<< sum(11,22,33); //calls func1
cout<<"sum="<< sum(10,15.5); //calls func2
cout<<"sum="<< sum(13.5,12.5); //calls func3
return 0;
}
int sum(int p,int q,int r){ //func1
 return(a+b+c);
 }
double sum(int l,double m){ //func2
 return(l+m);
 }
 float sum(float p,float q){ //func3
 return(p+q);
 }
```

# EXAMPLE : FUNCTION OVERLOADING

## // Program of function overloading when number of arguments vary.

```
#include <iostream>

class Cal {
    public:
static int add(int a,int b){
        return a + b;
    }
static int add(int a, int b, int c)
    {
        return a + b + c;
    }
};
int main(void) {
    Cal C;                                  //      class object declaration.
    cout<<C.add(10, 20)<<endl;
    cout<<C.add(12, 20, 23);
   return 0;
}
```

# EXAMPLE : FUNCTION OVERLOADING

## // Program of function overloading with different types of arguments.

```
#include<iostream>
using namespace std;
int mul(int,int);
float mul(float,int);


int mul(int a,int b)
{
    return a*b;
}
float mul(double x, int y)
{
    return x*y;
}
int main()
{
    int r1 = mul(6,7);
    float r2 = mul(0.2,3);
    std::cout << "r1 is : " <<r1<< std::endl;
    std::cout <<"r2 is : "  <<r2<< std::endl;
    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. Which of the following permits function overloading on c++?**

a) type

b) number of arguments

c) type & number of arguments

d) number of objects

## Multiple Choice Question:

**Q2. In which of the following we cannot overload the function?**

a) return function

b) caller

c) called function

d) main function

## Multiple Choice Question:

**Q3. Function overloading is also similar to which of the following?**

a) operator overloading

b) constructor overloading

c) destructor overloading

d) function overloading

## Multiple Choice Question:

**Q4. Overloaded functions are _____**

a) Very long functions that can hardly run

b) One function containing another one or more functions inside it

c) Two or more functions with the same name but different number of parameters or type

d) Very long functions

## Multiple Choice Question:

**Q5. What are the advantages of passing arguments by reference?**

a) Changes to parameter values within the function also affect the original arguments

b) There is need to copy parameter values (i.e. less memory used)

c) There is no need to call constructors for parameters (i.e. faster)

d) All of the mentioned

# Summary

## In this lecture, you learned that:

- Function overloading is the process of using the same name for two or more functions.
- Each redefinition of the function must use either different types of parameters or a different number of parameters.