RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

CSPS103: Object Oriented Programming

Lecture-25

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

preeti.ru@ramauniversity.ac.in

OBJECTIVES

In this lecture, you will learn to:

- ❖Operators Overloading
- ❖Syntax of Operator Overloading
- ❖Rules for Operator Overloading

OPERATOR OVERLOADING

- Operator overloading is a compile-time polymorphism in which the operator is overloaded to provide the special meaning to the user-defined data type.

- Operator overloading is used to overload or redefines most of the operators available in C++.

- It is used to perform the operation on the user-defined data type.

- For example, C++ provides the ability to add the variables of the user-defined data type that is applied to the built-in data types.

SYNTAX OF OPERATOR OVERLOADING

```
return_type class_name  : : operator op(argument_list)
{
    // body of the function.
}
```


- Where the **return type** is the type of value returned by the function.
- **class_name** is the name of the class.
- **operator op** is an operator function where op is the operator being overloaded, and the operator is the keyword.

RULES FOR OPERATOR OVERLOADING

- Existing operators can only be overloaded, but the new operators cannot be overloaded.
- The overloaded operator contains at least one operand of the user-defined data type.
- We cannot use friend function to overload certain operators. However, the member function can be used to overload those operators.
- When unary operators are overloaded through a member function take no explicit arguments, but, if they are overloaded by a friend function, takes one argument.
- When binary operators are overloaded through a member function takes one explicit argument, and if they are overloaded through a friend function takes two explicit arguments.

PROCESS OF OVERLOADING

The process of overloading involves the following steps:

- Create a class that defines the data type that is to be used in the overloading operation.
- Declare the operator function operator op() in the public part of the class.
- Define the operator function to implement the required operations.

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 binary operator?

a) Operator that performs its action on a single operand

b) Operator that performs its action on two operand

c) Operator that performs its action on three operand

d) Operator that performs its action on any number of operands

Multiple Choice Question:

Q2. Which is the correct example of a binary operator?

a) ++

b) —

c) Dereferencing operator(*)

d) +

Multiple Choice Question:

Q3. Which is the correct example of a unary operator?

a) &

b) ==

c) —

d) /

Multiple Choice Question:

Q4. Which is called ternary operator?

a) ?:

b) &&

c) |||

d) ===

Multiple Choice Question:

Q5. Which is the correct statement about operator overloading?

a) Only arithmetic operators can be overloaded

b) Only non-arithmetic operators can be overloaded

c) Precedence of operators are changed after overlaoding

d) Associativity and precedence of operators does not change

Summary

In this lecture, you learned that:

- The language allows not only functions to be overloaded, but also most of the operators, such as +, -, *, /, etc.
- As the name suggests, here the conventional operators can be programmed to carry out more complex operations.