

Inline Function and Function Overloading

Michelle Johnson*

¹Professor, Central University of Las Villas, Norway

INTRODUCTION

Inline function is used in program is to save some memory space which has become appreciable, when the function is likely to be called many times. Every time, a function is called it takes a lot of extra time in executing a series of instructions for a particular task. Such as jumping to the function, pushing arguments into the stack and so on and line function is a function that is expanding in line when it is invoked means when the compiler replaces the function call with the corresponding function code. It includes functions like

```
Int add (int a, int b) ;// prototype 1
```

```
Int add (int a, int b) ;// prototype 2
```

```
Double add (double x, double p) ;// prototype 3
```

```
Double add (int p, double q) ;// prototype 4
```

```
Double add (double p, int q) ;// prototype 5
```

Function call

```
cout << add (5, 10, 15);
```

```
cout << add (10, 7, 5, 6);
```

```
cout << add (5, 10)
```

Overloading means the use of the same thing for different purposes. Using Function Overloading we can design a family of functions with one function name but with different argument list or we can use the same function name to create function that perform a variety of tasks, it is also known as Function Polymorphism. An operator which exhibits more than one form is known as Operator Overloading or the mechanism of giving

special task to an operator is known as Operator Overloading. Overloading Unary Operator is a minus operator when used as a unary, it takes just one operand. This operator changes the sign of an operand when applied to an object, one should change the sign of each of its data objects.

There are some rules for Operator Overloading. Only existing operator can be overloaded, new operators cannot be overloaded. The overloaded operator must have at least user defined least one operand that is of user defined type. We cannot change the basic meaning of the operator. That is to say, we cannot redefine the plus (+) operator to subtract one value from the other. Overloaded operators follow the syntax rules of the original operators, they cannot be overridden. We cannot use the "friend" function to overload certain operator. However, member function can be used to overload them. Friend cannot be used with assignment operator (=), function call (), subscripting operator ([]), class member access operator (->).

Unary operator, overloaded by means of a member function, take no explicit arguments and return no explicit values, but those overloaded by means of a friend function, take one reference argument. Binary operators are overloaded through a member function, take one explicit argument and those which are overloaded through a friend function take two explicit arguments. When using binary operators are overloaded through a member function, the lefthand operand must be an object of the relevant class. Binary arithmetic operators such as +, -, *, / must explicitly return a value. They must not attempt to change their own arguments.

Correspondence to: Michelle Johnson, Professor, Professor, Central University of Las Villas, Norway, E-Mail Johnson.M@las.com

Received: July 05, 2021; **Accepted:** July 19, 2021; **Published:** July 26, 2021

Citation: Michelle J (2021) Inline Function and Function Overloading: Editorial. J Theor Comput Sci Open Access.7:4.

Copyright: © 2021 Michelle J. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.