C++

A S R Pavan
Scientist ‘B’
NIELIT Calicut

Topics to be discussed

- Operator overloading (type of polymorphism)
- Overloading assignment operator
- Operator overloading as member function
- Operator overloading as global function

Operator overloading

- Using the operators with user-defined types.
- user defined data types to behave similar to built-in types
- Majority of the c++ operators can be overloaded
- Operators which cannot be overloaded are
- Precedence and associavity are same
- Can overload an operator as its type only.
- Can’t create new operators
- [], (), -> and = operators must be declared as member methods

Can’t overload Operator
::
?:
.*
.
sizeof

Operator overloading

- Syntax:

  Return_type class_name::operator symbol(args)

  - Return_type is the return type of the function
  - Operator is a keyword
  - Symbol is the operator we want to overload
  - Args are arguments to the function

Overloading assignment operator

- Operator overloading as member function
- Operator overloading as global function

Operator overloading as member func.

- Member functions are overloading functions declared as members of a class.
- It gains access to all of the member variables and functions of that class.
- The overloaded function must be added as a member function of the left operand.
- The left operand becomes the implicit *this object
- All other operands become function parameters.

Operator overloading as member func.

- The object on the left hand side of the operator must be an object of the class we are using.

Operator overloading as global func.

- Global functions are not member functions.
- To access the private attributes, global functions often declared as friend functions of the class.
- unary operator: single object in the parameter list
- binary operator: two object in the parameter list
- Never use member and non-meber(global) overloading functions at a time because it confuses the compiler.

Doubts

Q&A

End of the session

Thank You