

Operator Overloading, friend operator and this pointers

1. Define a class Complex with appropriate instance variables and member functions. Overload following operators
 - a. << insertion operator
 - b. >> extraction operator
2. Define a class Complex with appropriate instance variables and member functions. One of the functions should be setData() to set the properties of the object. Make sure the names of formal arguments are the same as names of instance variables.
3. Overload subscript operator [] that will be useful when we want to check for an index out of bound.
4. Create a student class and overload new and delete operators as a member function of the class.
5. Create a student class and overload new and delete operators outside the class.
6. Create a complex class and overload assignment operator for that class.
7. Create an Integer class and overload logical not operator for that class.
8. Create a Coordinate class for 3 variables x,y and z and overload comma operator such that when you write c3 = (c1 , c2) then c2 is assigned to c3. Where c1,c2,and c3 are objects of 3D coordinate class.
9. Create an Integer class that contains int x as an instance variable and overload casting int() operator that will type cast your Integer class object to int data type.
10. Create a Distance class having 2 instance variable feet and inches. Also create default constructor and parameterized constructor takes 2 variables . Now overload () function call operator that takes 3 arguments a , b and c and set feet = a + c + 5 and inches = a+b + 15.
11. Create a class Marks that have one member variable marks and one member function that will print marks. We know that we can access member functions using (.) dot operator. Now you need to overload (->) arrow operator to access that function.