**CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY**

**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH**

Department of Computer Engineering/Computer Science & Engineering/ Information Technology

**Subject Name: Object Oriented Programming with C++**

**Semester: II**

**Subject Code: CE144**

**Academic year: 2020-21**

|  |  |
| --- | --- |
| **No.** | **Aim of the Practical** |
| **27.** | **Create a class Measure having members: meter and cm. The class has get( ) and put( ) functions. Overload operator + and – such that they support M1=M2+15 and M3=M1 – 4.5. Also overload + and – such that they support M1=5.0+M2 and M3=2.0 – M4. Write a main( ) to test the class.**  **Use the concept of Overloading Binary Operators with friend function.**  **PROGRAM CODE :**  #include <iostream>  using namespace std;  class Measure  {  private:  float meter, cm;  public:  void getdata()  {  cout << "Enter Meter: ";  cin >> meter;  cout << "Enter cm: ";  cin >> cm;  }  void putdata()  {  cout << "Meter is: " << meter << " and cm is: " << cm << endl;  cout << endl;  }  Measure operator+(float i)  {  Measure m;  m.meter = meter + i;  m.cm = cm + i;  return m;  }  Measure operator-(float i)  {  Measure m;  m.meter = meter - i;  m.cm = cm - i;  return m;  }  Measure friend operator+(float i, Measure m1);  Measure friend operator-(float i, Measure m1);  };  Measure operator+(float i, Measure m1)  {  Measure m;  m.meter = m1.meter + i;  m.cm = m1.cm + i;  return m;  }  Measure operator-(float i, Measure m1)  {  Measure m;  m.meter = m1.meter - i;  m.cm = m1.cm - i;  return m;  }  int main()  {  Measure m1, m2, m3;  m1.getdata();  m1.putdata();  cout << endl;  m2.getdata();  m2.putdata();  cout << "M1 = M2 + 15" << endl;  m1 = m2 + 15;  m1.putdata();  cout << "M3 - M1 - 4.5" << endl;  m3 = m1 - 4.5;  m3.putdata();  cout << "M1 = 5.0 + M2" << endl;  m1 = 5.0 + m2;  m1.putdata();  cout << "M3 = 2.0 - M2" << endl;  m3 = 2.0 - m2;  m3.putdata();  return 0;  }  **OUTPUT:**    **CONCLUSION:** In this practical we learnt how to use overloading binary operators with friend function. |