BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)
Dept. of CSE

Rupnagar, Mirpur-2, Dhaka-1216, Bangladesh
Phone: PABX-9024266, 9024277, 9015397, 9020132-4, Fax: 9024399
E-mail: info@bubt.edu.bd, Website: www.bubt.edu.bd

EXAMINATION SCRIPT

(Filled up by class teacher)

Question No.	Mark Obtained
Total Mark	

(This part is filled up by students)

Name:	Md. Zobayer Hasan Nayem
Student ID:	19202103274
Intake - Section:	44-7
Program:	CSE
Course code:	121
Course title:	CSE 121
Trimester:	Summer 2020
Exam type:	Class Test-2
Date:	1st November, 2020
Question option:	

Ans To the Que No : 01

Ans to the que NO: 01

Is it possible to overload Input output operator using member function, yes. Because the overload operator is defined as a member instead of a friend. An overloaded declaration is a declaration that is declared with the same name as a previously declared in the same scope. When we call an overloaded operator, The compiler determines the most appropriate definition to use by comparing the argument type you have used to call the function or operator with the parameter types specified in the definitions.

* There is a code :-

```
# include <iostream>
using namespace std;

class Distance {
  private:
    int feet;
    int inches;
  public:
    Distance() {
        feet = 0;
        inches = 0;
    }
    Distance (int f, int i) {
        feet = f;
        inches = i;
    }
    friend ostream &operator << (ostream &output,
                                 const Distance &D) {
        output << "F : " << D.feet << "I : " << D.inches;
        return output;
    }
```

```
friend istream &operator >> (istream &input, Distance &D){
        input >> D.feet >> D.inches;
        return input;
    }
};

int main() {
    Distance D1(11, 10), D2(5, 11), D3;
    cout << "Enter the value of object :" << endl;
    cin >> D3;
    cout << "First Distance : " << D1 << endl;
    cout << "Second Distance : " << D2 << endl;
    cout << "Third Distance : " << D3 << endl;

    return 0;
}
```