National University of Computer and Emerging Sciences 

**Manual**

for

Data Structures Lab 1

Lab Instructor(s)

Samia Akhter

Sonia

Section DS

| Semester | Fall 2022 |
| --- | --- |

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

In this lab, students will practice:

1. Templates

**Question 1**

a. Create a template function “compareTemp” that will work as a generic function to compare:  Int

 Float

 Char

bool compareTemp (T x, T y);

b. Create a partial specialized function “compareTempPtr” that will compare pointers to Object bool compareTemp (T\* x, T\* y);

c. Create a complete specialized template function “compareTemp” that will serve the purpose of string comparison.

bool compareTemp< const char\* > (const char\* x, const char\* y);

d. Test your implementation by using the following main function:

int main(int argc, char const \*argv[])

{

int x = 'a', y = 97;

char a = 'a', b = 'a';

char \*aptr = &a, \*bptr = &b;

string str1 = "String", str2 = "String";

(compareTemp(x, y)) ? cout << "Integers are equal" : cout << "Integers are not e qual";

cout << endl;

(compareTemp(a, b)) ? cout << "Chars are equal" : cout << "Chars are not equal"; cout << endl;

(compareTemp(aptr, bptr)) ? cout << "Pointers are equal" : cout << "Pointers are not equal";

cout << endl;

(compareTemp(str1, str2)) ? cout << "Strings are equal" : cout << "Strings are n ot equal";

return 0;

}

**Question 2**

a. Create a template class “DataObject” that will have the following data members:  T data  U key  DataObject\* Link b. Also implement setters and getters member functions for all the data members of this class:  void setData (T data)  T getData()

You can initialize the link member to a null pointer.

c. Also create constructor and destructor for this class

DataObject(); ~DataObject();

d. Overload operator “=” to deep copy of the object to a new object

DataObject< U, T > operator=( DataObject< U, T > const& obj)

e. Also overload operator “==” to compare the class objects

f. Test your implementation by using the following main function:

int main(int argc, char const \*argv[])

{

DataObject<int, char> Obj1 = DataObject<int, char>(97, 'a');

DataObject<int, char> Obj2 = DataObject<int, char>(Obj1);

DataObject<int, char> Obj3 = DataObject<int, char>(1, 'x');

(Obj1 == Obj2)) ? cout << "Objects are equal" : cout << "Objects are not equal";

Obj2.setLink(&Obj1);

Obj2.setData('b');

Obj2.setKey(98);

cout << endl;

cout << "[ " << Obj1.getData() << " : " << Obj1.getKey() << " : " << Obj1.getLink() << " ]";

Obj1.~DataObject();

cout << endl;

cout << "[ " << Obj2.getData() << " : " << Obj2.getKey() << " : " << Obj2.getLink() << " ]";

cout << endl;

cout << "[ " << Obj3.getData() << " : " << Obj3.getKey() << " : " << Obj3.getLink() << " ]";

Obj3 = Obj2;

Obj2.~DataObject();

cout << endl;

cout << "[ " << Obj3.getData() << " : " << Obj3.getKey() << " : " << Obj3.getLink() << " ]";

Obj3.~DataObject();

return 0;

}