**Name (s): Kaumil Patel**

**Course Name:** Principles of Software Design

**Lab Section:** B01

**Course Code:** ENSF 480

**Assignment Number:** Lab 2

**Submission Date and Time:** 9/30/2021

**Exercise A - Overloading Operators in C++ (16 marks)**

Printing list just after its creation ...

List is EMPTY.

Printing list after inserting 3 new keys ...

8001 Dilbert

8002 Alice

8003 Wally

Printing list after removing two keys and inserting PointyHair ...

8003 Wally

8004 PointyHair

Printing list after changing data for one of the keys ...

8003 Sam

8004 PointyHair

Printing list after inserting 2 more keys ...

8001 Allen

8002 Peter

8003 Sam

8004 PointyHair

\*\*\*----Finished dictionary tests---------------------------\*\*\*

Testig a few comparison and insertion operators.

Peter is greater than or equal Allen

Allen is less than Peter

Peter is not equal to Allen

Peter is greater than Allen

Peter is not less than Allen

Peter is not equal to Allen

Using square bracket [] to access elements of Mystring objects.

The socond element of Peter is: e

The socond element of Poter is: o

Using << to display key/datum pairs in a Dictionary list:

Allen

Peter

Sam

PointyHair

Using [] to display the datum only:

Allen

Peter

Sam

PointyHair

Using [] to display sequence of charaters in a datum:

A

l

l

e

n

\*\*\*----Finished tests for overloading operators ----------\*\*\*

Process finished with exit code 0

ostream &operator<<(ostream &os, const Mystring &str) {  
 os << str.charsM;  
 return os;  
}  
  
bool Mystring::operator>=(const Mystring &rhs) const {  
 if(this->lengthM > rhs.lengthM){  
 return true;  
 }  
 for(int i=0;i<this->lengthM;i++){  
 if(this->charsM[i]>rhs.charsM[i]){  
 return true;  
 }else if(this->charsM[i] < rhs.charsM[i]){  
 return false;  
 }  
 }  
 return true;  
}  
  
bool Mystring::operator<=(const Mystring &rhs) const {  
 if(this->lengthM < rhs.lengthM){  
 return true;  
 }  
 for(int i=0;i<this->lengthM;i++){  
 if(this->charsM[i]<rhs.charsM[i]){  
 return true;  
 }else if(this->charsM[i] > rhs.charsM[i]){  
 return false;  
 }  
 }  
 return true;  
}  
  
bool Mystring::operator>(const Mystring &rhs) const {  
 if(this->lengthM > rhs.lengthM){  
 return true;  
 }  
 for(int i=0;i<this->lengthM;i++){  
 if(this->charsM[i]>rhs.charsM[i]){  
 return true;  
 }else if(this->charsM[i] < rhs.charsM[i]){  
 return false;  
 }  
 }  
 return false;  
}  
  
bool Mystring::operator<(const Mystring &rhs) const {  
 if(this->lengthM < rhs.lengthM){  
 return true;  
 }  
 for(int i=0;i<this->lengthM;i++){  
 if(this->charsM[i]<rhs.charsM[i]){  
 return true;  
 }else if(this->charsM[i] > rhs.charsM[i]){  
 return false;  
 }  
 }  
 return false;  
}  
  
bool Mystring::operator==(const Mystring &rhs) const {  
 if(this->lengthM != rhs.lengthM){  
 return false;  
 }  
 for(int i=0;i<this->lengthM;i++){  
 if(this->charsM[i]!=rhs.charsM[i]){  
 return false;  
 }  
 }  
 return true;  
}  
  
bool Mystring::operator!=(const Mystring &rhs) const {  
 if(this->lengthM != rhs.lengthM){  
 return true;  
 }  
 for(int i=0;i<this->lengthM;i++){  
 if(this->charsM[i]!=rhs.charsM[i]){  
 return true;  
 }  
 }  
 return false;  
}  
  
char &Mystring::operator[](const int &index) const {  
 return charsM[index];  
}

ostream &operator<<(ostream &os, const DictionaryList &dl) {  
 Node \*curr = dl.headM;  
 while (curr != 0) {  
 os << curr->getData() << endl;  
 curr = curr->getNext();  
 }  
 return os;  
}  
  
Mystring &DictionaryList::operator[](const int &index) {  
 if (index >= sizeM) {  
 exit(-1);  
 }  
 Node \*curr = headM;  
 for (int i = 0; i < index; i++) {  
 curr = curr->nextM;  
 }  
 return curr->datumM;  
}

**Exercise B - Inheritance in C++ (20 marks)**

Expected to dispaly the distance between m and n is: 3

The distance between m and n is: 3

Expected second version of the distance function also print: 3

The distance between m and n is again: 3

Testing Functions in class Square:

Square Name: SQUARE - S

X-coordinate: 5.00

Y-coordinate: 7.00

Side a: 12.00

Area: 144.00

Perimeter: 48.00

Testing Functions in class Rectangle:Rectangle Name: RECTANGLE A

X-coordinate: 5.00

Y-coordinate: 7.00

Side a: 12.00

Side b: 15.00

Area: 180.00

Perimeter: 54.00

Rectangle Name: RECTANGLE B

X-coordinate: 16.00

Y-coordinate: 7.00

Side a: 8.00

Side b: 9.00

Area: 72.00

Perimeter: 34.00

Distance between square a, and b is: 11.00

Rectangle Name: RECTANGLE A

X-coordinate: 5.00

Y-coordinate: 7.00

Side a: 12.00

Side b: 15.00

Area: 180.00

Perimeter: 54.00

Testing assignment operator in class Rectangle:

Rectangle Name: RECTANGLE rec2

X-coordinate: 3.00

Y-coordinate: 4.00

Side a: 11.00

Side b: 7.00

Area: 77.00

Perimeter: 36.00

Expected to display the following values for objec rec2:

Rectangle Name: RECTANGLE A

X-coordinate: 5

Y-coordinate: 7

Side a: 12

Side b: 15

Area: 180

Perimeter: 54

If it doesn't there is a problem with your assignment operator.

Rectangle Name: RECTANGLE A

X-coordinate: 5.00

Y-coordinate: 7.00

Side a: 12.00

Side b: 15.00

Area: 180.00

Perimeter: 54.00

Testing copy constructor in class Rectangle:

Rectangle Name: RECTANGLE A

X-coordinate: 5.00

Y-coordinate: 7.00

Side a: 100.00

Side b: 200.00

Area: 20000.00

Perimeter: 600.00

Expected to display the following values for objec rec2:

Rectangle Name: RECTANGLE A

X-coordinate: 5

Y-coordinate: 7

Side a: 100

Side b: 200

Area: 20000

Perimeter: 600

If it doesn't there is a problem with your assignment operator.

Rectangle Name: RECTANGLE A

X-coordinate: 5.00

Y-coordinate: 7.00

Side a: 100.00

Side b: 200.00

Area: 20000.00

Perimeter: 600.00

Testing array of pointers and polymorphism:

Square Name: SQUARE - S

X-coordinate: 5.00

Y-coordinate: 7.00

Side a: 12.00

Area: 144.00

Perimeter: 48.00

Rectangle Name: RECTANGLE B

X-coordinate: 16.00

Y-coordinate: 7.00

Side a: 8.00

Side b: 9.00

Area: 72.00

Perimeter: 34.00

Rectangle Name: RECTANGLE A

X-coordinate: 5.00

Y-coordinate: 7.00

Side a: 12.00

Side b: 15.00

Area: 180.00

Perimeter: 54.00

Rectangle Name: RECTANGLE A

X-coordinate: 5.00

Y-coordinate: 7.00

Side a: 100.00

Side b: 200.00

Area: 20000.00

Perimeter: 600.00

Process finished with exit code 0