Name: Noor-ul-Ain Sagheer

Reg #: BCS223020

Date: 10th Nov, 2023

Course: "OOP"

Section: 1

Assignment No: '2'

Description: (Copy Constructors, file handling)

Submitted to: "Sir Adnan Jelani"

Capital University of Science & Technology

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

Question # 01:
(Part A)

Shallow Copy: (In case of arrays)

- It makes a copy of the main object but not the things inside it.
- So, changes in the copied object affect the original object and vice versa.
- **Real-life example:** Imagine it's like taking a photo of a group, and if someone in the group changes their shirt, it's reflected in the photo too.

Example: 1

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

class example
{
  Private:
        int age;
        string *name;
        int size;
  Public:
      example (int s, int a)
      {
```

```
    size = s;
    name = new string [size];
    age = a;
}

example (example & other) // Shalow Copy Constructor
                          //when dynamic memory
                          // is under use.
{
    size = other.size;
    name = other.name;
    age = other.age;
}
void set-data (int i, string n)
{
    if (i>=0 && i<size)
    {
        name [i] = n;
    }
}
void display()
{
    for (int i=0; i<size; i++)
    {
        cout<<"Names : "<<name [i]<<"  ";
        cout<<endl;
    }
    cout<<"Age : "<<age;
}
```

```
~example()
{
    delete [] name;
}
};
void main()
{
    example obj1(5, 19);        // original object
    obj1.set-data(0, "noor");
    obj1.set-data(1, "sara");

    obj1.set-data(2, "Laiba");
    example copy = obj1;        // creating copy

    copy.set-data(0, "hoor");   // change reflect in
                                   both objs

    obj1.display();
    copy.display();
}
```

Deep Copy:

- It makes a copy of the main object and everything inside it.
- Changes in the copied object donot effect the original object and vise versa.
- They are like two separate things.

- **Real-life example:** Think of it like making a duplicate of a group, including each person, so changes in one group donot affect the other.

Example 2: (keep in eye first example)

We can make a deep copy of objects by only following this step;

```
example(example & other) // deep copy Constructor
{
    size = other.size;
    name = new string (*(other.name)); // now
// here we did deep copy of objects
}
```

Part B:

Code:

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

 class course
{
  Private:
    String course-name;
    string *students-name;
    int count, size;
```

```
Public:
  course(int s, int c)
  {
     count = c;  size = s;
     students-name = new string[size];
  }
  course(course & other) //deep Copy Constructor
  {
    course-name = other.course-name;
    students-name = new string (*(other.students-name));
    count = other.count;
    size = other.size;
  }
   void set-data(string c-n)
   {
      course-name = c-n;
   }
   void add-students(int index, string name)
  {
     if(index >=0 && index <size)
     {
        students-name[index] = name;
        count++;
     }
  }
   void drop-student(int index)
  {
     if(index >=0 && index <count)
      {
         for(int i= index; i< count -1; i++)
```

```
    {
      students_name[i] = students_name[i+1];
    }
    count--;
  }
}
String COURSE_NAME()
{
  return course_name;
}
String *students()
{
  return students_name;
}
int number_of_students()
{
  return count;
}
~course()
{
  delete [] students_name;
}
};

void main()
{
  course obj(5,0);
obj.set_data("OOP");
obj.add_students(0,"noor");
obj.add_students(1,"Laiba");
```

```
obj.add-students(2, "sara");
obj.add-students(3, "Ali");
obj.add-students(4, "Asad");
obj.drop-student(1);

cout<< "Course name: " <<endl;
cout<< obj.COURSE-NAME() <<endl; // original object

String * stu = obj.students();
cout << "Students name :" <<endl;
for (int i=0; i<obj.number-of-students(); i++)
{
    cout << stu[i] <<endl;
}
int num = obj.number-of-students();
cout<<endl;
cout<<" Number of students : " << num <<endl;

course copy(obj);   // copied object

copy.set-data(" Data Structures");

copy.add-students(0, "saba");

copy.add-students(1, "Amna");

copy.add-students(2, "Maryam");

copy.drop-student(0);
```

```
cout<< "Second object data : "<<endl;
cout<< "Course name : "<< copy.COURSE-NAME()<<endl;
string * student = copy.Students();

cout<< "Students name : "<< endl;
for (int i=0; i< copy.number-of-students(); i++)
  {
     cout<< student[i]<<endl;
  }

int number = copy.number-of-students();
 cout<< "Number of Students : "<< number<<endl;
}
```

Question # 02:

```
#Include<iostream>
#Include<string>
#Include<fstream>
#Include<s stream>
 using namespace std;

  int main()
{
  ifstream reading("shoppinglist.txt");
  if(!reading.is-open())

  {
     cout<< "Error in opening a file!"<<endl;
     return 1;
  }
```

```
string Line;
double total_price = 0.0;

cout << "Product Details : " << endl;

while (getline (reading, line))
{
    stringstream ss(line);

    string product_name;
    double product_price;

    getline (ss, product_name, ',');
    ss >> product_price;

    cout << "Product = " << product_name << " , price = "
    << product_price << endl;

    total_price = total_price + product_price;
}

cout << "Total price = " << total_price << endl;
reading.close();

return 0;
}
```

Github Account details:

Github email: chazeem19800@gmail.com

Github ID: NOOR-UL-AIN-SAGHEER