Nick Krisulevicz

Dr. Wang

COSC 120-751

01/25/2021

Lab 9

Lab 9.1

Source Code:

```
//Nick Krisulevicz
//Lab 9.1
//01/25/2021

#include <iostream>
using namespace std;

class Square //class declaration
{
   public:
      Square(float num);
      Square();
      ~Square();
      float side;
      void setSide(float);
      float findArea();
      float findPerimeter();

};

int main()
{
```

```
        Square box;       // box is defined as an object of the Square class
        float size;       // size contains the length of a side of the square

        cout << "Please enter the length of sides of the square: ";
        cin >> size;
        cout << endl;

        box.setSide(size);

        cout << "The area of the square is: " << box.findArea() << endl;
        cout << "The perimeter of the square is: " << box.findPerimeter() << endl;

   Square box1(size); //box1 is defined

        cout << "The area of the square is: " << box1.findArea() << endl;
        cout << "The perimeter of the square is: " << box1.findPerimeter() << endl;

        return 0;
}

//member function implementation
void Square::setSide(float length)
{
        side = length;
}

float Square::findArea()
{
        return side * side;
```

```
}

float Square::findPerimeter()
{
        return 4 * side;
}

Square::Square(float num) //constructor with parameters
{
   num = 9;
   setSide(num);
}

Square::Square() //default constructor
{
   cout << "The default constructor has been invoked." << endl;
}

Square::~Square() //destructor
{
   cout << "The destructor has been invoked." << endl;
}
```

Question Answer:

Exercise 1: Wanted to complete the client and implementation code and record the output. The output is here:

```
Please enter the length of sides of the square: 8

The area of the square is: 64
```

The perimeter of the square is: 32

Exercise 2: Wanted to add a constructor with parameters, a default constructor, and a destructor. The output is here:

The default constructor has been invoked.

Please enter the length of sides of the square: 4

The area of the square is: 16

The perimeter of the square is: 16

The area of the square is: 81

The perimeter of the square is: 36

The destructor has been invoked.

The destructor has been invoked.

Lab 9.2

Source Code:

```
//Nick Krisulevicz
//Lab 9.2
//01/25/2021

#include <iostream>
using namespace std;

//class declaration
class Circles
{
public:
	void setCenter(int x, int y);
	double findArea();
	double findCircumference();
```

```
        void printCircleStats();
        Circles(float r);
        Circles(int x, int y);
        Circles(float r, int x, int y);
        Circles();
        ~Circles();
private:
        float     radius;
        int       center_x;
        int       center_y;
};

const double PI = 3.14;

//client

int main()
{
   int x;
   int y;
   int rad;

   cout << "Enter the X coordinate: ";
        cin >> x;
        cout << "Enter the y coordinate: ";
        cin >> y;
        cout << "Enter the radius: ";
        cin >> rad;
        cout << endl;
```

```
Circles sphere(rad, x, y);
	Circles sphere1(2);
	Circles sphere2;
	Circles sphere3(rad, 15, 16);

	sphere.printCircleStats();

	cout << "The area of the circle is " << sphere.findArea() << endl;
	cout << "The circumference of the circle is "
		<< sphere.findCircumference() << endl << endl;

sphere1.printCircleStats();

	cout << "The area of the circle is " << sphere1.findArea() << endl;
	cout << "The circumference of the circle is "
		<< sphere1.findCircumference() << endl << endl;

sphere2.printCircleStats();

	cout << "The area of the circle is " << sphere2.findArea() << endl;
	cout << "The circumference of the circle is "
		<< sphere2.findCircumference() << endl << endl;

sphere3.printCircleStats();

	cout << "The area of the circle is " << sphere3.findArea() << endl;
	cout << "The circumference of the circle is "
		<< sphere3.findCircumference() << endl << endl;
```

```
	return 0;
}

//implementation

Circles::Circles()
{
	radius = 1;
	setCenter(0, 0);
}

Circles::Circles(float r)
{
   radius = r;
   setCenter(9, 10);
}

Circles::Circles(int x, int y)
{
   radius = 1;
   setCenter(x, y);
}

Circles::Circles(float r, int x, int y)
{
   radius = r;
   setCenter(x, y);
}
```

```
double Circles::findArea()
{
   return PI * (radius * radius);
}

double Circles::findCircumference()
{
   return 2 * (PI * radius);
}

void Circles::printCircleStats()
{
        cout << "The radius of the circle is " << radius << endl;
        cout << "The center of the circle is (" << center_x
                << "'" << center_y << ")" << endl;
}

void Circles::setCenter(int x, int y)
{
        center_x = x;
        center_y = y;
}

Circles::~Circles()
{
   cout << "This concludes the Circles class." << endl;
}
```

Question Answer:

Exercise 1: Wanted to complete the code and change it to assign value to the circle's radius and center during initialization of the object. The output is as follows.

The radius of the circle is 8

The center of the circle is (9'10)

The area of the circle is 200.96

The circumference of the circle is 50.24

Exercise 2: Wanted to create more objects and more constructors. Sphere1 took a radius and default coordinates. Sphere2 took the default constructor values and there were three circle outputs. They are as follows.

Enter the X coordinate: 4

Enter the y coordinate: 5

Enter the radius: 6

The radius of the circle is 6

The center of the circle is (4'5)

The area of the circle is 113.04

The circumference of the circle is 37.68

The radius of the circle is 2

The center of the circle is (9'10)

The area of the circle is 12.56

The circumference of the circle is 12.56

The radius of the circle is 1

The center of the circle is (0'0)

The area of the circle is 3.14

The circumference of the circle is 6.28

Exercise 3: Wanted to add another object, sphere3 with a default radius and 15,16 as the center. The output is the same as above with the addition of output for one more circle.

The radius of the circle is 9

The center of the circle is (15'16)

The area of the circle is 254.34

The circumference of the circle is 56.52

Exercise 4: Wanted to add destructors for each object. It should print four times since we have four objects created. The output is the same as before but with the destructor messages.

This concludes the Circles class.

This concludes the Circles class.

This concludes the Circles class.

This concludes the Circles class.

Lab 9.3

Source Code:

```
//Nick Krisulevicz
//Lab 9.3
//01/25/2021

#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;

const int MAX_LENGTH = 50;
class FloatList

{
```

```
public:
        void getList(ifstream&);
        void printList() const;

   void listAverage();

        FloatList();
        ~FloatList();

private:
        int length;
        float values[MAX_LENGTH];

};

int main()
{
        ifstream tempData;

        FloatList templist;

        cout << fixed << showpoint;

        cout << setprecision(2);

        tempData.open("temperatures.txt");

        templist.getList(tempData);
        templist.printList();
```

```
        templist.listAverage();


        return 0;
}

FloatList::FloatList()
{
   length = 0;
}

void FloatList::getList(ifstream& tempData)
{
   while(!tempData.eof())
   {
      for (int i = 0; i < 5; i++)
      {
         tempData >> values[i];
      }
   }
}

void FloatList::printList() const
{
   for(int j = 0; j < 5; j++)
   {
      cout << values[j] << endl;
   }
}
```

```
void FloatList::listAverage()
{
   float total = 0;
   for(int k = 0; k < 5; k++)
   {
      total += values[k];
   }
   float average = total / 5;
   cout << "Average temperature is " << average << endl;
}

FloatList::~FloatList()
{
   cout << "Destructor has been activated" << endl;
}
```

Question Answer:

Exercise 1: Wanted us to answer why printList has a const after it, and getList does not. printList is probably constant because we don't want the list to be modified in any way by this function. getList is not constant because that function reads the data from the file and assigns it into the array.

Exercise 2: Wanted us to finish the code and run the program. The output is as follows.

78.90

87.40

60.80

70.40

75.60

Exercise 3: Wanted us to add another member function that calculated the average of the temperatures read into the array. The output is as follows. I added a destructor as well.

78.90

87.40

60.80

70.40

75.60

Average temperature is 74.62

Destructor has been activated

Lab 9.4

Source Code:

```
//Nick Krisulevicz
//Lab 9.4
//01/25/2021

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

//declaration

const int NUMOFPROD = 10;

class Inventory
{
public:
	void getId(int item);

	void getAmount(int num);

	void display();
```

```
private:
        int     itemNumber;
        int     numOfItem;

};

//client

int main()
{
   Inventory inv;
        ifstream infile;
        infile.open("Inventory.dat");

        int products[NUMOFPROD];
        int stock[NUMOFPROD];

        int pos;
        int id;
        int total;

        while(!infile.eof())
   {
      for (pos = 0; pos < NUMOFPROD; pos++)
      {
         infile >> products[pos];
         infile >> stock[pos];
```

```
            id = products[pos];
            total = stock[pos];

            inv.getId(id);
            inv.getAmount(total);

            inv.display();
        }
    }

    infile.close();

            return 0;
}

//implementation

void Inventory::getId(int item)
{
    itemNumber = item;
}

void Inventory::getAmount(int num)
{
    numOfItem = num;
}

void Inventory::display()
{
```

```
    cout << "Item number " << itemNumber << " has " << numOfItem << " in stock." << endl;
}
```

Question Answer:

Exercise 1: Wanted to complete the code and run the program. The output from the inventory.dat file is as follows.

Item number 986 has 8 in stock.

Item number 432 has 24 in stock.

Item number 132 has 100 in stock.

Item number 123 has 89 in stock.

Item number 329 has 50 in stock.

Item number 503 has 30 in stock.

Item number 783 has 78 in stock.

Item number 822 has 32 in stock.

Item number 233 has 56 in stock.

Item number 322 has 74 in stock.

Lab 9.5

Source Code:

```
//Nick Krisulevicz
//Lab 9.5
//01/25/2021

#include <iostream>
#include <iomanip>
#include <string>
using namespace std;

//declaration
```

```
class SavingsAccount
{
   private:
      int dollars;
      int cents;
   public:
      SavingsAccount(int d, int c);
      SavingsAccount();
      char openAccount(char choice);
      void deposit();
      void withdrawal();
      void showBalance();
};

//client

int main()
{
   cout << fixed << setprecision(2) << showpoint;

   char decision1;
   string decision2;
   int doll = 0;
   int cent = 0;
   cout << "Would you like to open a bank account? (Y/N) ";
   cin >> decision1;

   cout << "Please enter principal value of dollars: ";
   cin >> doll;
```

```
cout << endl;

cout << "Please enter principal value of cents: ";
cin >> cent;
cout << endl;

SavingsAccount bank1(doll, cent);
SavingsAccount bank2;

if(decision1 == 'Y' || decision1 == 'y')
{
   bank1.openAccount(decision1);
   bank2.openAccount(decision1);
   cout << "Would you like to deposit or withdraw? ";
   cin >> decision2;
   cout << endl;
   if(decision2 == "Deposit" || decision2 == "deposit")
   {
      cout << "Bank account 1" << endl;
      bank1.deposit();
      cout << "Bank account 2 (default constructor)" << endl;
      bank2.deposit();
      cout << "Bank account 1" << endl;
      bank1.showBalance();
      cout << "Bank account 2 (default constructor)" << endl;
      bank2.showBalance();
   }
   else if(decision2 == "Withdraw" || decision2 == "withdraw")
   {
```

```
        cout << "Bank account 1" << endl;
        bank1.withdrawal();
        cout << "Bank account 2 (default constructor)" << endl;
        bank2.withdrawal();
        cout << "Bank account 1" << endl;
        bank1.showBalance();
        cout << "Bank account 2 (default constructor)" << endl;
        bank2.showBalance();
    }
}
else
{
    cout << "Okay bye." << endl;
}

cout << "Would you like to perform another deposit or withdrawal? (Y/N) ";
cin >> decision1;
cout << endl;

if(decision1 == 'Y' || decision1 == 'y')
{
    cout << "Would you like to deposit or withdraw? ";
    cin >> decision2;
    cout << endl;
    if(decision2 == "Deposit" || decision2 == "deposit")
    {
        cout << "Bank account 1" << endl;
        bank1.deposit();
        cout << "Bank account 2 (default constructor)" << endl;
```

```
            bank2.deposit();
            cout << "Bank account 1" << endl;
            bank1.showBalance();
            cout << "Bank account 2 (default constructor)" << endl;
            bank2.showBalance();
        }
        else if(decision2 == "Withdraw" || decision2 == "withdraw")
        {
            cout << "Bank account 1" << endl;
            bank1.withdrawal();
            cout << "Bank account 2 (default constructor)" << endl;
            bank2.withdrawal();
            cout << "Bank account 1" << endl;
            bank1.showBalance();
            cout << "Bank account 2 (default constructor)" << endl;
            bank2.showBalance();
        }
    }
    else
    {
        cout << "Okay bye." << endl;
    }




    return 0;
}
```

```
//implementation

char SavingsAccount::openAccount(char choice)
{
   dollars = 0;
   cents = 0;
}

void SavingsAccount::deposit()
{
   int dollaradd;
   int centadd;
   cout << "Enter amount of dollars to deposit: ";
   cin >> dollaradd;
   cout << endl;

   cout << "Enter amount of cents to deposit: ";
   cin >> centadd;
   cout << endl;

   dollars += dollaradd;
   cents += centadd;
   if(cents >= 100)
   {
      dollars += 1;
      cents -=100;
   }
}
```

```
void SavingsAccount::withdrawal()
{
   int dollarsub;
   int centsub;
   cout << "Enter amount of dollars to withdraw: ";
   cin >> dollarsub;
   cout << endl;

   cout << "Enter amount of cents to withdraw: ";
   cin >> centsub;
   cout << endl;

   if(cents < 0)
   {
      dollars -= 1;
      cents +=100;
   }
}

void SavingsAccount::showBalance()
{
   if(cents <= 10)
   {
      cout << "Your current balance is: $" << dollars << ".0" << cents << endl;
   }
   else
   {
      cout << "Your current balance is: $" << dollars << "." << cents << endl;
   }
```

```
}


SavingsAccount::SavingsAccount()
{
   dollars = 0;
   cents = 0;
}


SavingsAccount::SavingsAccount(int d, int c)
{
   dollars = d;
   cents = c;
}
```

Question Answer:

Exercise 1: Wanted to write the code for a program with a class used to open a bank account, deposit, withdraw, and show the balance. The deposit and withdraw values are assigned by the user. The object is bank1. The output is as follows.

Would you like to open a bank account? (Y/N) y

Would you like to deposit or withdraw? deposit

Enter amount of dollars to deposit: 25

Enter amount of cents to deposit: 50

Your current balance is: $25.50

Would you like to perform another deposit or withdrawal? (Y/N) y

Would you like to deposit or withdraw? withdraw

Enter amount of dollars to withdraw: 25

Enter amount of cents to withdraw: 50

Your current balance is: $0.00

Exercise 2: Wanted to create a second object, bank2. Bank2 uses the default constructor and bank1 uses the parametered constructor. Here is the output.

Would you like to open a bank account? (Y/N) y

Please enter principal value of dollars: 13

Please enter principal value of cents: 37

Would you like to deposit or withdraw? deposit

Bank account 1

Enter amount of dollars to deposit: 20

Enter amount of cents to deposit: 44

Bank account 2 (default constructor)

Enter amount of dollars to deposit: 10

Enter amount of cents to deposit: 00

Bank account 1

Your current balance is: $20.44

Bank account 2 (default constructor)

Your current balance is: $10.00

Would you like to perform another deposit or withdrawal? (Y/N) n

Okay bye.