Sinai University

Faculty of Information Technology

Computer Programming (2) Quiz #1

10-11-2018 Time allowed: One Hour

Name: .... ID: ....

Answer the following Questions **[15 marks]**

Question 1: **[5 marks]**

Define a class **report** that has four data members: ***adno*** (of type **int**) , ***name*** (of type **string**), ***marks*** (array of 5 floating values), and ***average***(of type float) represent the average of 5 marks, and it has three member functions ***getavg()*** to compute the average obtained in five subject, ***readinfo()*** to accept values for adno, name, marks, then calculate the value of average, and ***displayinfo()*** to display all data members of report on the screen.

Question 2: **[10 marks]**

Define a class **account** that represents a bank account and has three data members: account number **accNo** (of type int), current balance **balance** (of type double), and periodic interest rate **rate** (of type double). It also has:

(a) a constructor that creates an account object and sets its data members to given values.

(b) a member function **show_account()** that displays the information of an account object.

(c) a member function **deposit()** that accepts an amount of money, then increases the balance of an account object by this amount.

(d) a member function **withdraw()** that accepts an amount of money, then decreases the balance of an account object by this amount if it does not exceed the balance and returns the amount withdrawn; otherwise, returns 0.

(e) a member function **compound()** that computes the interest of an account object and adds it to its balance, where interest = balance x rate.

Then, write a program that performs the following tasks:

(i) Reads the information of an accounts from the keyboard, and create **account** objects for these data, then

(ii) Asks the user to enter one of the following letters and performs the corresponding operation:

D: Asks the user to enter an account number and an amount of money, and deposits this amount in the specified account, then displays the account.

W: Asks the user to enter an account number and an amount of money, and attempts to withdraw this amount from the specified account, then displays a message indicating whether the withdrawal is done or not.

C: Computes the interest for the ٥ accounts, then displays the accounts

(iii) Display the data of the account object

With my best wishes

Sinai University

Faculty of Information Technology

Computer Programming (2) **Quiz #1**

10-11-2018 **Model Answer** **Time allowed: One Hour**

Answer the following Questions [15 marks]

Question 1: **[5 marks]**

Define a class **report** that has four data members: ***adno*** (of type **int**) , ***name*** (of type **string**), ***marks*** (array of 5 floating values), and ***average***(of type float) represent the average of 5 marks, and it has three member functions ***getavg()*** to compute the average obtained in five subject, ***readinfo()*** to accept values for adno, name, marks, then calculate the value of average, and ***displayinfo()*** to display all data members of report on the screen.

```
class report
{
private:
      int adno;
      char name[10];
      float marks[5], average;
public:
      void getavg()
      {
            float s = 0;
            for (int i = 0;  i < 5;  i++)
            {
                  s = s + marks[i];
                  average = s / 5;
            }
      }
      void readinfo()
      {
            cin >> adno >> name;
            for (int i = 0; i < 5; i++)
                  cin >> marks[i];
            getavg();
      }
      void displayinfo()
      {
            cout << adno << name;
            for (int i = 0; i < 5; i++)
                  cout << marks[i];
            cout << average;
      }
};
```

Question 2: **[10 marks]**

Define a class **account** that represents a bank account and has three data members: account number **accNo** (of type int), current balance **balance** (of type double), and periodic interest rate **rate** (of type double). It also has:

(a) a constructor that creates an account object and sets its data members to given values.

(b) a member function **show_account()** that displays the information of an account object.

(c) a member function **deposit()** that accepts an amount of money, then increases the balance of an account object by this amount.

(d) a member function **withdraw()** that accepts an amount of money, then decreases the balance of an account object by this amount if it does not exceed the balance and returns the amount withdrawn; otherwise, returns 0.

(e) a member function **compound()** that computes the interest of an account object and adds it to its balance, where interest = balance x rate.

Then, write a program that performs the following tasks:

(i) Reads the information of an accounts from the keyboard, and create **account** objects for these data, then

(ii) Asks the user to enter one of the following letters and performs the corresponding operation:

D: Asks the user to enter an account number and an amount of money, and deposits this amount in the specified account, then displays the account.

W: Asks the user to enter an account number and an amount of money, and attempts to withdraw this amount from the specified account, then displays a message indicating whether the withdrawal is done or not.

C: Computes the interest for the ٥ accounts, then displays the accounts

(iii) Display the data of the account object

```cpp
#include<iostream>
using namespace std;
class account
{
private:
	int accno;
	double balance, rate;
public:
	account(int n, double b, double r)
	{
		accno = n;
		balance = b;
		rate = r;
	}
	void show_account()
	{
		cout << accno << balance << rate;
	}
	void deposit(double m)
	{
		balance = balance + m;
	}
	double withdraw(double m)
	{
		if (balance >= m)
		{
			balance == balance - m;
			return m;
		}
		else
			return 0;
	}
	void compound()
	{
		double intrest = rate * balance;
		balance = balance + intrest;
	}
};
```

```
void main()
{
	int n;
	double b, r, m;
	char h;
	cin >> n >> b >> r;
	account  c1(n, b, r);
	cout << "enter your choice (D or W or C)";
	cin >> h;
	if (h == 'd')
	{
		cout << "enter the ammount of money";
		cin >> n>> m;
		c1.deposit(m);
		c1.show_account();
	}
	else if (h == 'w')
	{
		cout << "enter the ammount of money";
		cin >>n >>m;
		double am=c1.withdraw(m);
		if (am > 0)
			cout << "withdraw completed";
		else
			cout << "withdraw not completed";
	}
	else if (h == 'c')
	{
		c1.compound();
		c1.show_account();
	}
	c1.show_account();
}
```

With my best wishes

Sinai University

Faculty of Information Technology

Computer Programming (2)

Quiz #2

Time allowed: One Hour

1-12-2018

Name: .... ID: ....

Answer the following Questions **[15 marks]**

Question 1: **[5 marks]**

Define a class **Array**, whose objects are arrays of elements of type int. Class **Array** has two data members: *data* (array of int) that holds the elements of the array, and ***size*** (of type int) that holds the number of elements in the array. And class **Array** has the following functions:

- A **constructor,** which receives an int argument that represents the size of the **Array** object and initializes the array elements to zeros.
- **getSize()** that returns the size of an **Array** object.
- ***getelements()*** that read the elements of an array object from the keyboard.
- ***Showelements()*** that print the elements of an **Array** object on the screen.
- Operator = that copy the elements of a given **Array** object to another.
- **Operator = =** that compares two **Array** objects and returns true if they have the same elements.
- **Operator +** that sum the elements of a two given **Array** objects.

Using the class *Array*, write a main program that performs the following tasks:

1. Create three objects, ar1, ar2 and ar3 of the class Array.
2. Read the elements of the two Array: ar1 and ar2.
3. Compare the two **Array** objects ar1 and ar2 and display one the following messages accordingly:
   - The two arrays are identical.
   - The two arrays are not identical.
4. Sum the elements of two **Array** objects ar1 and ar2 in ar3.
5. Copy the elements of **Array** object ar2 to **Array** object ar1.
6. Display the elements of the **Array** object ar3 and ar1.

With my best wishes

SINAI UNIVERSITY

Sinai University

Faculty of Information Technology

Computer Programming (2) **Quiz #3**

22-12-2018 **Time allowed: One Hour**

Name: .... **ID: ....**

Answer the following Questions **[15 marks]**

Question 1: Show the output of the following program:

```
class account {
  protected:
    int acntNo;
    float balance;
  public:
    account(int n, float bal=100.0)
    { acntNo=n; balance=bal; }
    virtual void display()
    {cout << "\n" << acntNo << "," << balance; }
    float getBalance() { return balance; }
};
class savAcnt : public account {
  private:
    float rate;
  public:
    savAcnt(int n, float bal,
                    float rt): account(n, bal)
    { rate = rt; }
    void display()
    {cout << "\n" << acntNo << ","
          << balance << "," << rate; }
    float getBalance()
    { return balance += balance * rate; }
};

void main () {
  account* acnt[4];
  acnt[0]=new account(1234, 500.0);
  acnt[1]=new savAcnt(2468, 600.0, 0.10);
  acnt[2]=new savAcnt(3579, 400.0, 0.15);
  acnt[3]=new account(1235);
  for (int i=0; i<4;i++)
    acnt[i]->display();
  for (int i=0; i<4;i++)
    cout<<"\nbalanace:" << acnt[i]->getBalance();
}
```

Question 2

1. Create a base class *Circle* that has 3 data members, ***x***, ***y***, ***r*** and ***area***, representing the coordinates of the center, the radius, and the area of a circle, and has a constructor that initializes the data members of *Circle* objects with given values. It also has 2 member functions: (1) ***show_data()*** to display the values of the center coordinates and radius of a *Circle* object, and (2) ***getArea()*** that calculates the area of a *Circle* object by using the formula πr^2, where **π=3.14**.

2. Create a child class *Cylinder* that has an additional data member ***h*** representing the cylinder height, and has a constructor that initializes the data members of the *Cylinder* objects with given values. It redefines the member function ***show_data()*** to display the values of the data members of the *Cylinder* object, and redefines the member function ***getArea()*** to calculate the surface area of a *Cylinder* object by using the formula: $(2 \times \pi \times r^2) + (2 \times \pi \times r \times h)$

3. Write a main program that creates an array of 5 pointers to **Circle**. In a loop, gets from the user data about a circle or a cylinder, and use **new** to create an object of type **Circle** or *Cylinder* to hold the data, and then put the pointer to the created object in the array. When the user has finished entering the data for all figures, display the data for all figures entered.

With my best wishes

Sinai University

Faculty of Information Technology

Computer Programming (2) **Quiz #2**

1-12-2018 **Model Answer** **Time allowed: One Hour**

Name: .... **ID: ………….………..**

Answer the following Questions **[15 marks]**

Question 1: **[5 marks]**

Define a class **Array**, whose objects are arrays of elements of type int. Class **Array** has two data members: ***data*** (array of int) that holds the elements of the array, and ***size*** (of type int) that holds the number of elements in the array. And class **Array** has the following functions:

- A **constructor,** which receives an int argument that represents the size of the **Array** object and initializes the array elements to zeros.
- **getSize()** that returns the size of an **Array** object.
- ***getelements()*** that read the elements of an array object from the keyboard.
- ***Showelements()*** that print the elements of an **Array** object on the screen.
- Operator = that copy the elements of a given **Array** object to another.
- **Operator = =** that compares two **Array** objects and returns true if they have the same elements.
- **Operator +** that sum the elements of a two given **Array** objects.

Using the class ***Array***, write a main program that performs the following tasks:

1. Create three objects, ar1, ar2 and ar3 of the class Array.
2. Read the elements of the two Array: ar1 and ar2.
3. Compare the two **Array** objects ar1 and ar2 and display one the following messages accordingly:
    - The two arrays are identical.
    - The two arrays are not identical.
4. Sum the elements of two **Array** objects ar1 and ar2 in ar3.
5. Copy the elements of **Array** object ar2 to **Array** object ar1.
6. Display the elements of the **Array** object ar3 and ar1.

```
#include<iostream>
using namespace std;
class Array
{
private:
      int data[100];
      int size;
public:
      Array(int s)
      {
            size = s;
            for (int i = 0; i < size; i++)
                  data[i] = 0;
      }
      int getsize()
      {
            return size;
      }
      void getelements()
      {
            for (int i = 0; i < size; i++)
                  cin >> data[i];
      }
      void showelements()
      {
            for (int i = 0; i < size; i++)
                  cout << data[i];
      }
      void operator =(Array a1)
      {
                  for (int i = 0; i < a1.size; i++)
                        data[i] = a1.data[i];
                  size = a1.getsize();
      }
      bool operator ==(Array a1)
      {
            bool ident = true;
                  for (int i = 0; i < size; i++)
                  {
                        if (data[i] != a1.data[i])
                        {
                              ident = false;
                              break;
                        }
                  }
                  return ident;
      }
Array operator + (Array a1)
      {
            Array a3(10);
            for (int i = 0; i < size; i++)
            {
                  a3.data[i] = data[i] + a1.data[i];
            }
            a3.size = size;
            return a3;
      }
};

void main()
{
      Array ar1(10), ar2(10), ar3(10);
      ar1.getelements();
      ar2.getelements();
      if (ar1 == ar2)
            cout << "The two arrays are identical.";
      else
            cout << "The two arrays are not identical.";
      ar3 = ar1 + ar2;
      ar1 = ar2;
      ar3.showelements();
      ar1.showelements();
}
```

Sinai University

Faculty of Information Technology

Computer Programming (2) **Quiz #2**

16-11-2019 **Time allowed: One Hour**

Model Answer

Answer the following Questions [15 marks]

Question 1:

Show the output of the following program:

```
class Circle {
   protected:
      int x, y, r;
   public:
     Circle(int xVal, int yVal, int rVal)
     { x=xVal;  y=yVal; r=rVal;}
     void display()
     {cout << x << ",“ << y << "," << r; }
     virtual float getArea()
      { return 3.14 * r * r; }
};
class Cylinder  : public Circle {
   private:
     int height;
   public:
     Cylinder(int xVal, int yVal, int rVal, int h)
                    : Circle(xVal, yVal, rVal)
     { height = h; }
     void display()
     {  cout << x << ",“  << y << ","
           << r << ",“ << height; }
       float getArea()
       { return 2 * 3.14 * r * height; }
};
void Show(Circle* shape)
{ cout << typeid(*shape).name() << ": "; }

void main () {
  Circle* shape[4];
  shape[0]=new Circle(3, 5, 10);
  shape[1]=new Cylinder(6, 8, 5, 20);
  shape[2]=new Circle(7, 9, 15);
  shape[3]=new Cylinder(1, 3, 7, 5);
  for (int i=0; i<4;i++)
  {
    Show(shape[i]);
    shape[i]->display();
    cout<<"\nArea:" << shape[i] -> getArea();
  }
}
```

```
3, 5,10
Area:314
6, 8,5
Area:628
7, 9,15
Area:706.5
1, 3,7
Area:219.8
```

Question 2:

Declare and implement the following class hierarchy that includes 4 classes, each one has a **constructor** that creates an object of the class and initializes its data members with given values:

A) A base class ***account*** that represents a bank account and has one data member **balance** that represents the current balance, and 3 member functions: **deposit()** that accepts an amount of money, then increases the balance by this amount, **get_balance()** that displays the balance, and **withdraw()** that accepts an amount of money, then decreases the balance by this amount if it does not exceed the balance and returns the amount withdrawn; otherwise, returns 0.

B) A child class ***curnt_acnt*** that represents a current account and has 2 additional data members: **limit** representing a lower limit for check cashing free of charge, and **charge** representing charge per check for lower balance, and redefines the member function **withdraw()** such that it accepts an amount of money **amt**, then decreases the balance by **amt**+**charge**, if the balance is less than **limit** and the value of **amt**+**charge** does not exceed the balance, and returns the amount withdrawn, otherwise it performs the withdrawal as in class ***account***.

C) A child class ***sav_acnt*** that represents a savings account and has an additional data member: **intrstRate** representing annual interest rate, and a member function: **compute_int()** that computes the interest, adds it to the balance and returns the interest, where interest = balance x rate.

D) A grandchild class ***special_acnt***, which represents a special account that combines the properties of the current account and the savings account, and has no data members of its own. It redefines the member function **withdraw ()** to perform the withdrawal as in class ***curnt_acnt***, and redefines **compute_int()** to compute the interest, add it to the balance and return the interest, only if the balance exceeds **limit**, otherwise it returns 0.

E) write a main program that performs the following tasks:

(i) Reads the information of accounts from the keyboard, and create ***special_acnt*** objects for these accounts, then

(ii) Asks the user to enter one of the following letters and performs the corresponding operation:

D: Asks the user to enter an amount of money, and deposits this amount in the specified account, then displays the account balance.

W: Asks the user to enter an amount of money, and attempts to withdraw this amount from the specified account, then displays a message indicating whether the withdrawal is done or not.

C: Computes the interest for the account, then displays the account balance.

```
#include <iostream>
#include <cmath>
using namespace std;
class account
{
protected:
	double balance;
public:
	account(double b)
	{
		balance = b;
	}
	void deposit(double amt)
	{
		balance = balance + amt;
	}
	void getbalance()
	{
		cout << balance;
	}
	double withdraw(double amt)
	{
		if (amt <= balance)
		{
			balance = balance - amt;
			return amt;
		}
		else
			return 0.0;
	}
};
class curnt_acnt :virtual public account
{
protected:
	double limit, charge;
public:
	curnt_acnt(double b, double lim, double ch) :account(b)
	{
		limit = lim;
		charge = ch;
	}
	double withdraw(double amt)
	{
		if (balance<limit && amt + charge>balance)
		{
			balance = balance - (amt + charge);
			return amt;
		}
		else
			return account::withdraw(amt);
	}
};
```

```
class sav_acnt :virtual public account
{
protected:
	double interstRate;
public:
	sav_acnt(double b, double rate) :account(b)
	{
		interstRate = rate;
	}
	double compute_int()
	{
		double interest = balance * interstRate;
		balance = balance + interest;
		return interest;
	}
};
class special_acnt : public curnt_acnt, public sav_acnt
{
public:
	special_acnt(double b, double lim, double ch, double rate)
		:account(b),curnt_acnt(b, lim, ch), sav_acnt(b, rate)
	{}
	double withdraw(double amt)
	{
		return curnt_acnt::withdraw(amt);
	}
	double compute_int()
	{
		if (balance > limit)
		{
			return sav_acnt::compute_int();
		}
		else
			return 0.0;
	}
};
```

```
int main()
{
	double b, ch, lim, rate,amt;
	char choice;
	cin >> b >> ch >> lim >> rate;
	special_acnt s1(b, lim, ch, rate);
	cout << "enter your choice (D,W,C)";
	cin >> choice;
	if (choice == 'D')
	{
		cout << "enter the amount of money";
		cin >> amt;
		s1.deposit(amt);
		s1.getbalance();
	}
	else if (choice == 'W')
	{
		cout << "enter the amount of money";
		cin >> amt;
		double w = s1.withdraw(amt);
		if (w == 0)
			cout << "the operation failed";
		else
			cout << "the operation succesful";
	}
	else if (choice == 'C')
	{
		double out=s1.compute_int();
		s1.getbalance();

	}
}
```

With my best wishes

Faculty of Information Technology

Final Exam of Computer Programming 2 (CSW 234)

Academic Year: 2018/2019
Semester: Fall 2018
Time: 3 hours

Level: 2nd Level
Date: 19-01-2019

(Total: 60 marks)

Answer about the following questions

Question 1: **[15 Marks]**

Declare and implement a class ***fraction*** that has two data members: ***num*** and ***denom***, (both of type **int**), which represent the fraction's numerator and denominator, respectively. It has two member functions: *getFract()* that reads the ***num*** and ***denom*** of a ***fraction*** object from the keyboard in fractional form (i.e. num/denom), and ***showFract()*** that displays a ***fraction*** object in fractional form. It also has two overloaded operators + and *, for performing the addition and multiplication operations on any two objects of class ***fraction***, respectively. Note that, the formulas for adding and multiplying two fractions a/b and c/d are as follows:

$$\frac{a}{b}+\frac{c}{d}=\frac{a*d+b*c}{b*d}, \qquad \frac{a}{b}\times\frac{c}{d}=\frac{a*c}{b*d}$$

Write a program that declares two objects of class ***fraction***, and repeatedly reads data for these two objects from the keyboard, then perform the two operations on them, displays the results, and asks the user whether he/she wants to continue or not.

Question 2: **[15 Marks]**

Define a base class **Person** that has two data members: ***name*** (char array) and ***idno*** (int). From this class derive two classes: (1) **Student**, which has an additional data member ***grade*** (char); and (2) **Professor**, which has two additional data members: ***rank*** (char array) and ***nPublications*** (int). Each of the three classes should have two member functions: ***getdata()*** to get its data from the keyboard, and ***showdata()*** to display its data.

Then, write a main program that creates an array of 10 pointers to **Person**. In a loop, gets from the user data about a student or a professor, and use **new** to create an object of type **Student** or **Professor** to hold the data, and then put the pointer to the created object in the array. When the user finishes entering the data for all persons, display the data for all persons entered.

[15 Marks]

Question 3:

Define a class *counter*, which has three data members: ***count*** (of type int) that represents the value of the counter, ***overflow*** (of type bool) that is set to *true* if an overflow occurs, and ***maxcnt*** (of type int) that represents the maximum value of the counter. It has a constructor ***counter****(int max_count)*, that creates a counter with a given maximum count and an initial value 0, and sets *overflow* to *false*, and it has 4 member functions: ***increment()***, which increases the current count by 1; but if the current count equals the maximum count, the *count* is set to zero, and *overflow* to true; ***reset()***, which sets the counter to zero and overflow to false; *get_count()*, which returns the current count, and *check_for_overflow()*, which returns true if an overflow has occurred and false if it has not.

Write a main program that reads a sentence, and counts the frequency of each vowel (a, e, i, o, u) in it, then displays these frequencies. The program should use a separate counter for each vowel, and it should display an error message if any of the counter's overflows.

[15 Marks]

Question 4:

A complex number consists of two parts: a real part and an imaginary part, and takes the form: (real_part) + i (imaginary_part). Two of the operations that can be performed on any two complex numbers, u = a + i b and v = c + i d, are:

Addition: u + v = (a+c) + i (b+d), Multiplication: u * v = (ac-bd) + i (ad+bc).

Declare and implement a class ***Complex*** that has: two data members of type float: *real* and *imag*, representing the real and imaginary parts of a complex number, and a **constructor** with no arguments that initializes the real and imaginary parts of an object of this class with zeros. It also has operator functions (+ and *) to perform the above operations on objects of this class, an **extraction operator >>**, that reads a ***Complex*** object in the form a+ib, an **insertion operator <<**, that displays a ***Complex*** object in the same form.

Using this class, write a main program that declares two ***Complex*** objects, asks the user to enter data for the two objects and an operator (+ or *), and performs the required operation, then displays the result.

With My Best Wishes

Faculty of Information Technology

Final Exam of Computer Programming 2 (CSW 234)

Academic Year: 2017/2018
Level: 2nd Level **Semester: Fall 2017**
Date: 23-12-2017 **Time: 3 hours**

<u>Answer about the following questions</u> **(Total: 60 marks)**

Question 1: **[15 Marks]**

A) Define the following object-oriented terms:

(i) ***Abstract base class***,
a class exists only to act as a parent of derived classes that will be used to instantiate objects

(ii) ***Pure virtual function.***
A pure virtual function is one with the expression =0

B) Show the output of the following program:

```
class Point {
   protected:
      int x, y;
   public:
     Point(int xx=0, int yy=0)
     { x = xx;  y = yy; }
     virtual void display()
    {cout << "(" << x << ",“ << y << ")"; }
     virtual float getArea()
     { return 0.0; }
};

class Circle : public Point {
   private:
     float radius;
   public:
     Circle(int xx, int yy, float r): Point(xx, yy)
    { radius = r; }
    void display()
    {cout << "(" << x << ",“ << y
           << ",“ << radius << ")"; }
       float getArea()
       { return 3.14*radius*radius; }
};
bool  hasArea(Point* p)
{ Circle* c;
  c = dynamic_cast<Circle*>(p);
  return c ? true : false;
}
void Show(Point* shape)
{ cout << "\n"<<typeid(*shape).name() << ": "; }

void main () {
  Point* List[4];
  List [0]=new Circle (2, 6, 10.0);
  List [1]=new Point (4, 5);
  List [2]=new Circle (3, 4, 20.0);
  List [3]=new Point;
  for (int i=0; i<4;i++) {
    show(List [i])
    List [i]->display();
    if ( hasArea(List[i]) )
      cout << "area=“ << List [i]-> getArea();
   }
}
```

```
class Circle: (2, 6, 10)area = 314
class Point: (4, 5)
class Circle: (3, 4, 20)area = 1256
class Point: (0, 0)
```

Question 2: **[15 Marks]**

A) <u>Declare and implement</u> a class ***fraction*** that has two data members: ***num*** and ***denom***, (both of type **int**), which represent the fraction’s numerator and denominator, respectively. It has two member functions: a 2-argument **constructor** that creates a ***fraction*** object and initializes its data members to given values***, an extraction operator*** **>>** that reads the ***num*** and ***denom*** of a ***fraction*** object from the keyboard in fractional form (i.e. num/denom), and **an insertion operator <<** that displays a ***fraction*** object in fractional form. It also has two overloaded operators **+** and *****, for performing the addition and multiplication operations on any two objects of class ***fraction***, respectively. Note that, the formulas for adding and multiplying two fractions a/b and c/d are as follows:

B) Add an exception class, and throw an exception in the 2-argument constructor and the member function ***extraction operator*** >> if the entered value of ***denom*** is zero. Use an argument in the exception constructor to report where the error has occurred.

C) Write a program that declares two objects of class ***fraction***, and repeatedly reads data for these two objects from the keyboard, then perform the two operations on them, displays the results, and asks the user whether he/she wants to continue or not.

```
#include<iostream>
#include<string>
using namespace std;
class fraction{
      int num, denom;
public:
      class excpnumzero{
      private:
            string  msg;
      public:
            excpnumzero(string s)
            { msg = s;   }
            void show_msg()
            { cout << msg;      }
      };
      fraction(int n, int d)
      {
            num = n;
            if (denom == 0)
                  throw excpnumzero("constructor error");
            denom = d;
      }
      friend istream& operator >>(istream & is, fraction &f1)
      {
            char h;
            is >> f1.num >> h >> f1.denom;
            if (f1.denom == 0)
                  throw excpnumzero("operator error");
            return is;
      }
      friend ostream & operator <<(ostream & os, fraction & f2)
      {
            os << f2.num << "/" << f2.denom;
            return os;
      }
fraction operator +(fraction f2)
      {
            fraction f3(0,1);
            f3.num = num*f2.denom + denom*f2.num;
            f3.denom = denom * f2.denom;
            return f3;
      }
      fraction operator *(fraction f2)
      {
            fraction f3(0, 1);
            f3.num = num*f2.num;
            f3.denom = denom * f2.denom;
            return f3;
      }
};
void  main()
{
      try
      {
            fraction f1(0, 1);
            fraction f2(0, 1);
            fraction f3(0, 1);
            cin >> f1;
            cin >>  f2;
            f3 = f1 + f2;
            cout << f3;
            f3 = f1*f2;
            cout << f3;

      }
catch (fraction::excpnumzero ex1)
      {
            ex1.show_msg();
      }
      system("pause");
}
```

Question 3: **[15 Marks]**

A) <u>Declare and implement</u> a class ***Date*** that has 3 **int** data members: day, month and year, and has a no-argument **constructor** that creates a ***Date*** object and initializes its data members to zeros, a 3-argument **constructor** that creates a ***Date*** object and initializes its data members to given values, and two member functions: ***get_date()*** that reads from the keyboard the data of a ***date*** object in the form dd/mm/yy, and ***show_date()*** that displays a ***date*** object in the form dd/mm/yy.

B) <u>Declare and implement</u> a class ***Person*** that has 2 data members: **name** and **birthdate** (an object of class ***date***, defined in **a**), and has two member functions: ***get_person()*** that reads from the keyboard the data of a ***Person*** object, and ***show_person()*** that displays the data of a ***Person*** object.

C) write a main program that reads a 10 of persons from the keyboard, and store them in a file called "pesons.dat". Then, the program reads the persons from the file and displays their information.

```
#include<iostream>
#include<string>
#include<fstream>
using namespace std;
class date{
private:
      int day, month, year;
public:
      date()
      {
            day = 0;     month = 0;   year = 0;    }
      date(int d, int m, int y)
      {
            day = d;     month = m;   year = y;    }
      void get_data()
      {
            char h;
            cin >> day >> h >> month >> h >> year;
      }
      void show_data()
      {
            cout << day << "/" << month << "/" << year;
      }
      friend istream& operator >>(istream & is, date &d1)
      {
            char h;
            is >> d1.day >> h >> d1.month >> h >> d1.year;
            return is;  }
      friend ostream & operator <<(ostream & os, date & d2)
      {
            os << d2.day << "/" << d2.month << "/" << d2.year;
            return os;  }
};
class person{
      string name;
      date birthdate;
public:
      void get_person()
      {
            cin >> name; birthdate.get_data();     }
      void show_person()
      {
            cout << name;      birthdate.show_data();    }
      friend istream& operator >>(istream & is, person &p1)
      {
            is >> p1.name;
            is >> p1.birthdate;
            return is;
      }
friend ostream & operator <<(ostream & os, person & p2)
      {           os << p2.name  << p2.birthdate;  return os;   }

};
```

```
void main()
{
	fstream outintf("person.dat", ios::in);
	person p1;
	for (int i = 0; i < 10; i++)
	{
		p1.get_person();
		outintf << p1;
	}
	outintf.close();
	outintf.open("person.dat", ios::out);
	for (int i = 0; i < 10; i++)
	{
		outintf >> p1;
		p1.show_person();
	}
	outintf.close();
	system("pause");
}
```

Question 4: **[15 Marks]**

A. Define a base class **Employee** that has two data members ***name*** and ***idno***. From this class derive two classes: **SalaryEmployee**, which adds a data member ***salary***; and **TempEmployee**, which adds two data members: ***hourlyPay*** and ***hoursWorked***. Each of the three classes should have two member functions: ***Getdata()*** to get its data from the user, and ***Showdata()*** to display its data.

B. Write a program that creates an array of 10 pointers to **Employee**. In a loop, gets from the user data about a salary employee or a temporary employee, and use **new** to create an object of type **SalaryEmployee** or **TempEmployee** to hold the data, and then put the pointer to the created object in the array. When the user has finished entering the data for all employees, display the data for all employees entered.

```
#include<iostream>
#include<string>
#include<fstream>
using namespace std;
class employee
{
protected:
	string name;
	int idno;
public :
virtual	void getdata()
	{	cin >> name >> idno;	}
virtual	void showdata()
	{	cout << name << idno; }
};
```

```
class salaryemployee:public employee{
private:
	double salary;
public:
	void getedata()
	{
		employee::getdata();
		cin >> salary;
	}
	void showdata()
	{
		employee::showdata();
		cout << salary;
	}
};

class tempemployee :public employee
{
private:
	double hourlypay, hoursworked;
public:
	void getdata()
	{
		employee::getdata();
		cin >> hourlypay >> hoursworked;
	}
	void showdata()
	{
		employee::showdata();
		cout << hourlypay << hoursworked;
	}

};

void main()
{
	char h;
	employee *e1[10];
	for (int i = 0; i < 10; i++)
	{
		cout << "enter your choice (S,T)";
		cin >> h;
		if (h == 's')
			e1[i] = new salaryemployee();
		else
			e1[i] = new tempemployee();
		e1[i]->getdata();
	}
	for (int i = 0; i < 10; i++)
	{
		e1[i]->showdata();
	}
}
```

Sinai University
Faculty of IT & CS
Course Title: Computer Programming(1)
Course Code: CSW 232

Date: 23-8-2021
Marks: 20
Time Allowed: 60 minutes
Semester: Summer 2021

Mid-Term Exam

A. Each of the following definitions and program segments has errors. Locate these errors:

Question	**Answer**

1.
int table[10];
for (int x = 0; x < 20; x++)
{
    cout << "Enter the next value: ";
    cin >> table[x];
}
	int table[10];
	for (int x = 0; x < 10; x++)
	{
	    cout << "Enter the next value: ";
	    cin >> table[x];
	}

2. int hours[3] = 8, 12, 16;	int hours[3] = {8, 12, 16} ;
[illegible] **3.** int numbers[8] = {1, 2, , 4, , 5};	int numbers[2] = {1, 2};
4. float ratings [ ];	float ratings [any integer literal];

5.
char greeting [ ] = {'H', 'e', 'l', 'l', 'o'};
cout << greeting;
	char greeting [ ] = {'H', 'e', 'l', 'l', 'o'};
	cout << greeting;//Erroe:using array name

6.
int array1[4], array2[4] = {3, 6, 9, 12};
array1 = array2;
	int array1[4], array2[4] = {3, 6, 9, 12};
	array1 = array2;

7.
void showValues(int nums)
{
    for (int count = 0; count < 8; count++)
    cout << nums[count];
}
	void showValues(int nums [])
	{
	    for (int count = 0; count < 8; count++)
	    cout << nums[count];
	}

8.
void showValues (int nums[4][ ])
{
    for (rows = 0; rows < 4; rows++)
    for (cols = 0; cols < 5; cols++)
    cout << nums[rows][cols];
}
	void showValues (int nums[ ][4])
	//Error: no of columns must be included in parameter list
	
	{
	    for (rows = 0; rows < 4; rows++)
	    for (cols = 0; cols < 5; cols++)
	    cout << nums[rows][cols];
	}

B.

1. Create a structure ***account*** that represents a bank account, and includes 2 members: ***accNo*** (account number of type int) and ***balance*** (current balance of type double).

```
struct account
{
        int accNo;
        double balance;
};
```

---

2. Write a function **get_account()** that reads from the keyboard the information of an account and stores them in a structure variable of type **account**, then returns it.

```
account get_account ()
{
        account a1;
        cou t<< "\n enter the number of account and its balance";
        cin >> a1.accNo >> a1.balance;
        return a1;
}
```

---

3. Write a function **deposit()** that accepts a structure variable of type **account** and an amount of money, then increases the balance by this amount.

```
void deposit (account a1, double m)
{
            a1.balance  +=  m;
 }
```

C. Write a class named **Car** that has the following member variables:

- **year**. An int that holds the car's year model.
- **make**. A string that holds the make of the car.
- **speed**. An int that holds the car's current speed.

In addition, the class should have the following constructor and other member functions.

- **Constructor**. The constructor should accept the car's year model and make as arguments. These values should be assigned to the object's year and make member variables. The constructor should also assign 0 to the speed member variables.
- **Accessor**. Appropriate accessor functions to get the values stored in an object's year, make, and speed member variables.
- **accelerate**. The accelerate function should add 5 to the speed member variable each time it is called.
- **brake**. The brake function should subtract 5 from the speed member variable each time it is called.

```
class car
{
private:
    int year;
    stirng make;
    int speed;
public:
    car(int y, string m)
    {
        year = y;
        speed = 0;
        make = m;
    }

    }
    void get_data(){

        cout << year << speed << make;
    }
    int getspeed()
    {
        return speed;
    }
    void accelerate()
    {
        speed += 5;
    }
    void brake()
    {
        speed -= 5;
    }
};
```

With my best wishes
Dr. Gaber Hassan

Sinai University
Faculty of IT & CS
Course Title: Computer Programming(2)
Course Code: CSW 234

Date: 23-11-2021
Marks: 20
Time Allowed: 60 minutes
Semester: Fall 2021

Mid -Term Exam

1) What will be the output of the following C++ codes?

```
#include <iostream>
using namespace std;
struct Time
{
   int hours, minutes, seconds;
};
int toSeconds(Time now);
int main()
{
   Time t;
   t.hours = 5;
   t.minutes = 30;
   t.seconds = 45;
   cout <<"Total  seconds  :  "  <<  toSeconds(t)  <<
     endl;
   return 0;
}
int toSeconds (Time now)
{
 return  3600  *  now.hours  +  60  *  now.minutes  +
     now.seconds;
}
```

Answer

```
Total seconds : 19845
```

```
#include <iostream>
using namespace std;
int main()
{
   // initialize an array without specifying size
   double numbers[] = {7, 5, 6, 12, 35, 27};
   double sum = 0;
   double count = 0;
   double average;
   cout << "The numbers are: ";
   //  print array elements
   for (int i = 0 ; i<6 ; i++)
   {
      cout << numbers [i] << "  ";
      //  calculate the sum
      sum += numbers [i];
      // count the no. of array elements
      ++count;
   }
   // print the sum
   cout << "\nTheir Sum = " << sum << endl;
   // find the average
   average = sum / count;
   cout << "Their Average = " << average << endl;
   return 0;
}
```

Answer

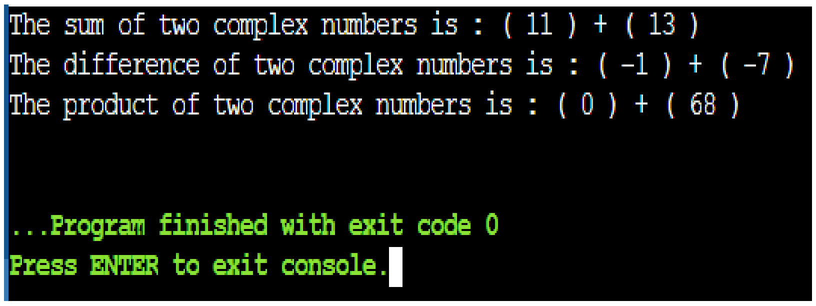
```
The numbers are: 7  5  6  12  35  27
Their Sum = 92
Their Average = 15.3333
```

2) **Create an array that can hold ten integers, and get input from user. Display those values on the screen, and then prompt the user for an integer. Search through the array, and count the number of times the item is found.**

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

int count(const int arr1[10], int numb);

int main()
{
    int arr[10];int num;
    cout<<"Enter 10 values in array:";

    for(int i = 0; i < 10; i++)
      cin >> arr[i];

    cout << "Values in array are now:";

    for(int i = 0; i < 10; i++)
      cout << " " << arr[i];

    cout<<"\n"<< "Enter value to find : ";
    cin >> num;

          cout<< num<< " was found " << count(arr,num);

          return 0;
}
int count(const int arr1[10], int numb)
{
 int cont=0;

   for (int u=0; u<10; u++)
   {

    if (arr1[u]==numb)
     cont=cont+1;
   }
 return cont;
}
```

3) Write a program to add, subtract and multiply two complex numbers using structures to function.

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

struct Complex
{
   int real;
   int imag;
};
Complex Add (Complex , Complex );
Complex Subtract (Complex , Complex );
Complex Multiply (Complex , Complex );
int main()
{
   Complex num1 = {5,3};
   Complex num2 = {6,10};
   Complex S, D, P;
   S = Add(num1,num2);
   D = Subtract(num1,num2);
   P = Multiply(num1,num2);
   cout<<"The sum of two complex numbers is : "<<"( "<<S.real<<" )"<<" + "<<"( "<<S.imag<<" )"<<endl;
   cout<<"The difference of two complex numbers is : "<<"( "<<D.real<<" )"<<" + "<<"( "<<D.imag<<" )"<<endl;
   cout<<"The product of two complex numbers is : "<<"( "<<P.real<<" )"<<" + "<<"( "<<P.imag<<" )"<<endl;
         return 0;
}
Complex Add (Complex a, Complex b)
 {
   Complex sum;
   sum.real = a.real + b.real;
   sum.imag = a.imag + b.imag;
   return sum;
 }
 Complex Subtract (Complex a, Complex b)
 {
   Complex sum;
   sum.real = a.real - b.real;
   sum.imag = a.imag - b.imag;
   return sum;  }
 Complex Multiply (Complex a, Complex b)
 {
   Complex sum;
   sum.real = (a.real * b.real) - (a.imag * b.imag);
   sum.imag = (a.real * b.imag) + (a.imag * b.real);
   return sum;
 }
```



```
The sum of two complex numbers is : ( 11 ) + ( 13 )
The difference of two complex numbers is : ( -1 ) + ( -7 )
The product of two complex numbers is : ( 0 ) + ( 68 )


...Program finished with exit code 0
Press ENTER to exit console.
```

4) Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' with the constructor having the three sides as its parameters.

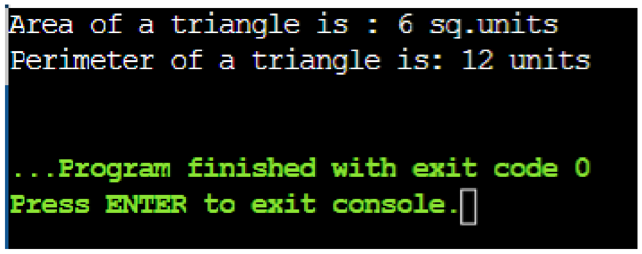
Hint:

Hero formula for calculating area and perimeter of rectangle is

$$Area = \sqrt{s(s-a)(s-b)(s-c)}, \quad where\ s = \frac{a+b+c}{2}, \qquad \text{perimeter} = a+b+c$$

$$a, b\ and\ c\ are\ the\ lengths\ of\ the\ three\ sides\ of\ the\ triangle$$

```
#include<iostream>
#include<cmath>
using namespace std;
class Triangle
{
private:
   int S1, S2, S3;
public:
   Triangle (int a, int b, int c)
{
    S1 = a;
    S2 = b;
    S3 = c;
}
   void area ( );
   void perimeter ( );
};
void Triangle::area ( )
{
   float S, A;
   S =((S1+ S2+ S3)/2.0);
   A=sqrt(S*(S- S1)*(S- S2)*(S- S3));
   cout<<"Area of a triangle is: "<<A<<" sq.units\n";
}
void Triangle::perimeter( )
{
    cout<<"Perimeter of a triangle is: "<< S1+ S2+ S3<<" units\n";
}
int main()
{
    Triangle tri1(3, 4, 5);
    tri1.area( );
    tri1.perimeter( );
    return 0;
}
```



```
Area of a triangle is : 6 sq.units
Perimeter of a triangle is: 12 units


...Program finished with exit code 0
Press ENTER to exit console.
```

With my best wishes

Dr. Gaber Hassan

Answer all the following questions

Question No: 1 [15 points]

A. What will the following program segments display?

1)

```
#include <iostream>
using namespace std;
// Function prototypes
void fillArray(char [], int);
void showArray(const char [], int);
int main ()
{
  const int SIZE = 8;
  char prodCode[SIZE] = {'0','0','0','0','0','0','0','0'};
  fillArray(prodCode, SIZE);
  showArray(prodCode, SIZE);
  return 0;
}
// Definition of function fillArray.
// (Hint: 65 is the ASCII code for 'A')
void fillArray(char arr[], int size)
{
  char code = 65;
  for (int k = 0; k < size; code++, k++)
  arr[k] = code;
}
// Definition of function showArray.
void showArray(const char codes[], int size)
{
for (int k = 0; k < size; k++)
  cout << codes[k];
  cout << endl;
}
```

2)

```
#include <iostream>
#include <string>
using namespace std;
class Base
{
  public:
  Base(){cout << "Entering the base.\n";}
  Base(string str)
  {
  cout << "This base is " << str << ".\n";
  }
  ~Base() {cout << "Leaving the base.\n";}
};
class Camp : public Base
{
  public:
  Camp(){cout << "Entering the camp.\n";}
  Camp(string str1, string str2) :
  Base(str1)
  {
  cout << "The camp is " << str2 << ".\n";
  }
};
int main()
{
  Camp outpost("secure", "secluded");
  return 0;
}
```

B. Write a program that lets the user enter 10 values into an array. The program should then display the largest and smallest values stored in the array.

Question No: 2 [15 points]

A. Customer Accounts

Write a program that uses a structure to store the following data about a customer account:

Name
Address
City
Telephone Number
Account Balance

The program should use an array of at least 10 structures. It should let the user enter data into the array, change the contents of any element, and display all the data stored in the array.

Input Validation: When the data for a new account is entered, ~~be sure the user enters data for~~ all the ~~fields~~. No negative account balances should be entered.

privete

B. PersonData and CustomerData classes

Design a class named PersonData with the following member variables:

- lastName
- firstName
- address
- city
- phone

Write the appropriate accessor and mutator functions for these member variables. Next, design a class named CustomerData , which is derived from the PersonData class. The CustomerData class should have the following member variables:

customerNumber
mailingList

The customerNumber variable will be used to hold a unique integer for each customer. The mailingList variable should be a bool . It will be set to true if the customer wishes to be on a mailing list, or false if the customer does not wish to be on a mailing list. Write appropriate accessor and mutator functions for these member variables. Demonstrate an object of the CustomerData class in a simple program.

Question No:3 [15 points]

A. Circle Class

Write a Circle class that has the following member variables:

- radius : a double
- pi : a double initialized with the value 3.14159

pr

The class should have the following member functions:

- **Default Constructor.** A default constructor that sets radius to 0.0.
- **Constructor.** Accepts the radius of the circle as an argument.
- **setRadius.** A mutator function for the radius variable.
- **getRadius.** An accessor function for the radius variable.
- **getArea.** Returns the area of the circle, which is calculated as area = pi * radius * radius
- **getDiameter.** Returns the diameter of the circle, which is calculated as diameter = radius * 2
- **getCircumference.** Returns the circumference of the circle, which is calculated as circumference=2*pi*radius

Write a program that demonstrates the Circle class by asking the user for the circle's radius, creating a Circle object, and then reporting the circle's area, diameter, and circumference.

Question No:4 [15 points]

A. **Declare and implement** a class ***fraction*** that has two data members: ***num*** and ***denom***, (both of type int), which represent the fraction's numerator and denominator, respectively. It has two member functions: ***getFract()*** that reads the ***num*** and ***denom*** of a ***fraction*** object from the keyboard in fractional form (i.e. num/denom), and ***showFract()*** that displays a ***fraction*** object in fractional form. It also has two overloaded operators + and *, for performing the addition and multiplication operations on any two objects of class ***fraction***, respectively. Note that, the formulas for adding and multiplying two fractions a/b and c/d are as follows:

$$\frac{a}{b}+\frac{c}{d}=\frac{a*d+b*c}{b*d}, \qquad \frac{a}{b}\times\frac{c}{d}=\frac{a*c}{b*d}$$

(a) **Write a program** that declares two objects of class ***fraction***, and repeatedly reads data for these two objects from the keyboard, then perform the two operations on them, displays the results, and asks the user whether he/she wants to continue or not.

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

void fillarray (char arr[], int size) {
      char code = 65;
      for (int k=0;k<size;code++,k++)
      arr [k] = code;
}

void showarray (const char codes[], int size){
      for (int k=0;k<size;k++)
      cout << codes[k];
      cout << endl;
}

int main () {
      const int SIZE = 8;
      char prodCode [SIZE] = {'0','0','0','0','0','0','0','0'};
      fillarray (prodCode, SIZE);
      showarray (prodCode, SIZE);
}
```

```
#include <iostream>
#include <string>

using namespace std;

class report {
      private:
            int adno;
            string name;
            float marks [5];
            float average;
      public:
            void getavg () {
                  int sum = 0;
                  for (int i = 0;i<5;i++) {
                        sum = sum + marks [i];
                  }
                  average = sum / 5;
            }

            void readinfo () {
                  cout << "Enter the adno: ";
                  cin >> adno;
                  cin.ignore();
                  cout << "Enter the name: ";
                  getline(cin,name);
                  for (int k=0;k<5;k++) {
                        cout << "Enter the mark number " << k+1 << ": ";
                        cin >> marks [k];
                  }
                  getavg ();
            }

            void displayInfo () {
                  cout << "\nThe adno is " << adno;
                  cout << "\nThe name is " << name;
                  cout << "\nThe average is " << average;
            }
};

int main () {
      report R1;
      R1.readinfo();
      R1.displayInfo();
}
```

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

struct Customer {
    string name, address, city, telephone;
    int balance;
};

int main()
{
    int i, counter = 1, count, validateBalance;
    const int SIZE = 10;
    Customer C1[SIZE];

    for (i = 0; i < SIZE; i++) {
        cout << "\nCustomer #" << counter << ": \n";
        cout << "Enter the name: ";
        getline(cin, C1[i].name);
        cout << "Enter the address: ";
        getline(cin, C1[i].address);
        cout << "Enter the city: ";
        getline(cin, C1[i].city);
        cout << "Enter the telephone: ";
        getline(cin, C1[i].telephone);
        cout << "Enter the balance (the amount should be a positive or
zero): ";
        cin.ignore();
        while (validateBalance < -1) {
            cin >> validateBalance;
        }
        C1[i].balance = validateBalance;

        counter++;
    }
}
```

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

class PersonData {
private:
    string lastName, firstName, address, city, phone;

public:
    PersonData()
    {
        lastName = firstName = address = city = phone = "";
    }

    // Non-default constructor
    PersonData(string ln, string fn, string ad, string ci, string ph)
    {
        lastName = ln;
        firstName = fn;
        address = ad;
        city = ci;
        phone = ph;
    }

    // Mutator functions prototypes:
    void setLastName();
    void setFirstName();
    void setAddress();
    void setCity();
    void setPhone();
    void setAllPerson();

    // Accessor functions:
    string getLastName() { return lastName; }
    string getFirstName() { return firstName; }
    string getAddress() { return address; }
    string getCity() { return city; }
    string getPhone() { return phone; }

    // Call all print functions:
    void printAllPerson()
    {
        cout << "Firstname is " << getLastName() << endl;
        cout << "Lastname is " << getFirstName() << endl;
        cout << "The address is " << getAddress() << endl;
        cout << "City of the address is " << getCity() << endl;
        cout << "The phone is " << getPhone() << endl;
    }
};

// Persondata - Mutators:
void PersonData::setLastName()
{
    cout << "Enter lastname: ";
    getline(cin, lastName);
}

void PersonData::setFirstName()
{
    cout << "Enter firstname: ";
    getline(cin, firstName);
```

```
}

void PersonData::setAddress()
{
    cout << "Enter address: ";
    getline(cin, address);
}

void PersonData::setCity()
{
    cout << "Enter city: ";
    getline(cin, city);
}

void PersonData::setPhone()
{
    cout << "Enter phone: ";
    getline(cin, phone);
}

void PersonData::setAllPerson()
{
    setLastName();
    setFirstName();
    setAddress();
    setCity();
    setPhone();
}

class CustomerData : public PersonData {
private:
    static int customerNumber;
    bool mailingList;

public:
    // Non-defaut constructor
    CustomerData(string ln, string fn, string ad, string ci, string ph,
bool ml)
        : PersonData(ln, fn, ad, ci, ph)
    {
        customerNumber++;
        mailingList = ml;
    }

    // Mutator functions:
    void setMailingList()
    {
        cout << "\nDo you want to subscribe to our mailing list?";
        cout << "Enter 1 for Yes, and enter 0 for no: ";
        cin >> mailingList;
    }

    void setAllCustomer()
    {
        PersonData::setAllPerson();
        setMailingList();
    }
    // Accessor functions;
    int getCustomerNumber() { return customerNumber; }
    int getMailingList() { return mailingList; }
    // Call all print functions:
    void printAllCustomer()
    {
        cout << "Customer number is " << getCustomerNumber() << endl;
        PersonData::printAllPerson();
        cout << "Mailing list subscription: " << getMailingList() <<
endl;
    }
};

int CustomerData::customerNumber = 0;

int main()
{
    CustomerData C1 = { "Mamoun", "Mohammad", "19, Downtown", "Cairo",
"019647150", 1 };
    C1.printAllCustomer();
    cout << endl << endl;
    CustomerData C2 = { "Diaa", "David", "80, Commerce", "New York",
"018019878", 0 };
    C2.printAllCustomer();
}
```

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

class Circle {
      private:
            double radius;
            const double pi = 3.14159;
      public:
            Circle () {
                  radius = 0.0;
            }

            Circle (double r) {
                  radius = r;
            }

            // Mutator functions:
            void setRadius () {
                  cout << "Enter the radius for the circle: ";
                  cin >> radius;
            }

            // Accessor functions:
            double getRadius () { return radius; }
            double getArea () { return (pi * radius * radius); }
            double getDiameter () { return (radius*2); }
            double getCircumference () { return (2*pi*radius); }
            void printAll () {
                  cout << "Radius of the circle is " << getRadius () <<
endl;
                  cout << "Area of the circle is " << getArea () << endl;
                  cout << "Diameter of the circle is " << getDiameter ()
<< endl;
                  cout << "Circumference of the circle is " <<
getCircumference() << endl;
            }
};

int main () {
      Circle C1;
      C1.setRadius();
      C1.printAll();
}
```

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

class fraction {
private:
    int num, denom;

public:
    fraction()
    {
        num = 0;
        denom = 0;
    }

    fraction(int n, int d)
    {
        num = n;
        denom = d;
    }

    void getFract()
    {
        cout << "Please enter the numerator: ";
        cin >> num;
        cout << "Please enter the denominator: ";
        cin >> denom;
    }

    void showFract()
    {
        cout << num << "/" << denom << endl;
        cout << "Which equals: " << double(num) / denom << endl;
    }

    fraction operator+(fraction object2)
    {
        fraction objectAdd;
        objectAdd.num = (num * object2.denom) + (denom * object2.num);
        objectAdd.denom = (denom) * (object2.denom);
        return objectAdd;
    }

    fraction operator*(fraction object2)
    {
        fraction objectMulti;
        objectMulti.num = (num) * (object2.num);
        objectMulti.denom = (denom) * (object2.denom);
        return objectMulti;
    }
};

int main()
{
    string result;
    fraction ob1, ob2, ob3;

    do {
        ob1.getFract();
        ob2.getFract();
        ob3 = ob1 + ob2;
        cout << "Sum is: ";
        ob3.showFract();
        cout << endl;
        cin.ignore();
        cout << "Do you want to continue? (Enter \"no\" or \'n\' to
exit): ";
        getline(cin, result);
    } while (result != "N" && result != "no" && result != "No" && result
!= "nO");
}
```