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) <u>Add</u> 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) <u>Write</u> 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) Declare and implement 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) Declare and implement 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();
        }
}
```