OOP LAB

Chetan Gupta

DTU/2K15/CO/044

INDEX

```cpp
//Create a class of a right angled triangle and calculate its Area and Perimeter
#include <iostream>
#include <math.h>

using namespace std;

class Triangle
{
    int base, height;
public:
    void set_triangle()
    {
        cout << "Base : ";
        cin >> base;
        cout << "Height : ";
        cin >> height;
    }
    float get_area()
    {
        return 0.5*base*height;
    }
    float get_perimeter()
    {
        return base + height + sqrt(base*base + height*height);
    }
};

int main()
{
    Triangle t;
    t.set_triangle();
    cout << "Area : " << t.get_area() << endl;
    cout << "Perimeter : " << t.get_perimeter() << endl;
}
```

```
OUTPUT:
Base : 3
Height : 4
Area : 6
Perimeter : 12
```

```cpp
//Create a class of complex numbers
#include <iostream>
using namespace std;
class Complex
{
    float real, imag;
public:
    Complex(float r, float i)
    {
        real = r;
        imag = i;
    }
    Complex add(Complex b)
    {
        Complex ans(0,0);
        ans.real = real + b.real;
        ans.imag = imag + b.imag;
        return ans;
    }
    Complex subtract(Complex b)
    {
        Complex ans(0,0);
        ans.real = real - b.real;
        ans.imag = imag - b.imag;
        return ans;
    }
    void print()
    {
        cout << real << " + " << imag << "i" << endl;
    }
};

int main()
{
    Complex a(1,2), b(3,4);
    cout << "A : ";
    a.print();
    cout << "B : ";
    b.print();
    cout << "A+B : ";
    (a.add(b)).print();
    cout << "A-B : ";
    (a.subtract(b)).print();
}

OUTPUT:
A : 1 + 2i
B : 3 + 4i
A+B : 4 + 6i
A-B : -2 + -2i
```

```cpp
//Using classes calculate taxes and employee salary
#include "iostream"
#include "iomanip"

using namespace std;

class Income
{
public:
    virtual double pay_salary()=0;
    virtual double compute_deductions()=0;
    virtual double calculate_tax()=0;
};

class Income2 : public Income
{
    double basic;
    double DA;
public:
    void set_salary()
    {
        basic = 1000000;
        DA = 50000;
    }
    double compute_deductions()
    {
        return 0.08*basic;
    }

    double pay_salary()
    {
        return (basic + DA + 0.15*basic - compute_deductions());
    }
    double calculate_tax()
    {
        double salary = pay_salary();
        if(salary <= 100000)
            return 0.2*salary;
        else
            return (0.3*salary + 0.3*0.1*salary);
    }
};

int main()
{
    Income2 obj;
    obj.set_salary();
    cout << fixed << setprecision(2);
    cout << "Deductions : " << obj.compute_deductions() << endl;
    cout << "Salary : " << obj.pay_salary() << endl;
    cout << "Tax : " << obj.calculate_tax() << endl;
}

OUTPUT:
Deductions : 80000.00
Salary : 1120000.00
Tax : 369600.00
```

```cpp
//Using classes maintain Employee details and increments
#include <iostream>
using namespace std;
class Employee
{
    long number;
    char dob[9];
    int rank;
    float salary;
public:
    Employee()
    {
        number=rank=salary=0;
    }
    void enter_details()
    {
        cout << "Number : ";
        cin >> number;
        cout << "DOB(dd-mm-yy) : ";
        cin >> dob;
        cout << "Starting Salary : ";
        cin >> salary;
    }
    void increment_salary(bool check=0)
    {
        if(check)
            salary = salary + 0.25*salary;
        else
            salary = salary + 0.10*salary;
    }
    void increment_rank()
    {
        rank++;
        increment_salary(1);
    }
    void print_details()
    {
        cout << "---------------------------\n";
        cout << "Number : " << number << endl;
        cout << "DOB : " << dob << endl;
        cout << "Rank : " << rank << endl;
        cout << "Salary : " << salary << endl;
        cout << "---------------------------\n";
    }
};

int main()
{
    Employee emp;
    emp.enter_details();
    cout << "After promoting rank : " << endl << endl;
    emp.increment_rank();
    emp.print_details();

}
```

```
OUTPUT:
Number : 12312
DOB(dd-mm-yy) : 19-09-97
Starting Salary : 100923
After promoting rank :

---------------------------
Number : 12312
DOB : 19-09-97
Rank : 1
Salary : 126154
---------------------------
```

```cpp
//Bank account management using classes
#include <iostream>
using namespace std;

class Account
{
    char name[50];
    long account_no;
    int account_type;
    double balance;
public:
    void new_acc()
    {
        cout << "Name : ";
        cin >> name;
        cout << "Account Number : ";
        cin >> account_no;
        cout << "Account Type : ";
        cin >> account_type;
        cout << "Starting Balance : ";
        cin >> balance;
    }
    void get_type()
    {
        if(account_type == 0)
        {
            cout << "Savings";
        }
        else
        {
            cout << "Current";
        }
    }
    void get_statement()
    {
        cout << "\n=======================================\n";
        cout << "Account Number : " << account_no << endl;
        cout << "Holder : " << name << endl;
        cout << "Account Type : ";
        get_type();
        cout << endl;
        cout << "Balance : " << balance << endl;
        cout << "\n=======================================\n";
    }
    void deposit()
    {
        double amount;
        cout << "Enter amount to deposit : ";
        cin >> amount;
        balance+=amount;
    }
    void withdraw()
    {
        double amount;
        cout << "Enter amount to withdraw : ";
        cin >> amount;
        if(balance > amount)
            balance-=amount;
        else
            cout << "Not Enough balance";
    }

};

int main()
{
    Account a;
    a.new_acc();
    a.deposit();
```

```
    a.get_statement();
    a.withdraw();
    a.get_statement();
}
```

```
OUTPUT:
Name : Chetan
Account Number : 123193129
Account Type : 0
Starting Balance : 100000
Enter amount to deposit : 123

========================================
Account Number : 123193129
Holder : Chetan
Account Type : Savings
Balance : 100123

========================================
Enter amount to withdraw : 123

========================================
Account Number : 123193129
Holder : Chetan
Account Type : Savings
Balance : 100000

========================================
```

```cpp
//Do operator overloading for string manipulation
#include <iostream>
using namespace std;

class Str
{
    char value[1000];
    int length;
public:
    Str()
    {
        length = 0;
    }
    Str(int n)
    {
        length = n;
        cin >> value;
    }
    Str operator + (Str b)
    {
        Str c;
        c.length = length + b.length;
        for(int i=0 ; i<length ; i++)
            c.value[i] = value[i];

        for(int i=length ; i<length+b.length ; i++)
            c.value[i] = b.value[i-length];
        return c;
    }
    Str operator = (Str b)
    {
        Str c;
        c.length = b.length;
        for (int i = 0; i < c.length; ++i)
            c.value[i] = b.value[i];
        return c;
    }
    bool operator < (Str b)
    {
        for(int i=0 ; i<min(length, b.length) ; i++)
        {
            if(value[i] > b.value[i])
                return false;
        }
        if(b.length < length)
            return false;
        else
            return true;
    }
    bool operator > (Str b)
    {
        for(int i=0 ; i<min(length, b.length) ; i++)
        {
            if(value[i] < b.value[i])
                return false;
        }
        if(b.length > length)
            return false;
        else
            return true;
    }
    bool operator == (Str b)
    {
        if(length != b.length)
            return false;
        for(int i=0 ; i<(length) ; i++)
        {
            if(value[i] != b.value[i])
                return false;
```

```
        }
        return true;
    }
    void print()
    {
        cout << value << endl;
    }

};

int main()
{
    Str a(3),b(4);
    Str c = a + b;
    cout << int(a < b) << endl;
    cout << int(a > c) << endl;
    cout << int(a == a) << endl;
}
```

OUTPUT:

```
abc
def
1
0
1
```

```cpp
// Using inheritance make product and manufacturer classes
#include <iostream>

using namespace std;

class Product
{
    long serial;
    char name[50];
public:
    Product()
    {
        cout << "Enter Product Details : \n";
        cout << "Serial : ";
        cin >> serial;
        cout << "Product Name : ";
        cin >> name;
    }
    void show_product()
    {
        cout << "Product Serial : " << serial << endl;
        cout << "Product Name : " << name << endl;
    }
};

class Manufacturer
{
    char name[50];
    char state[50];
public:
    Manufacturer()
    {
        cout << "Enter Manufacturer Details : \n";
        cout << "Name : ";
        cin >> name;
        cout << "State : ";
        cin >> state;
    }
    void show_manufacturer()
    {
        cout << "Manufacturer Name : " << name << endl;
        cout << "State : " << state << endl;
    }
};

class NCDT : public Product,public Manufacturer
{
public:
    void display()
    {
        cout << "------------------------------\n";
        show_product();
        show_manufacturer();
        cout << "------------------------------\n";
    }
};

int main()
{
    NCDT a;
    cout << endl;
    a.display();
}

OUPUT:
Enter Product Details :
Serial : 12312
Product Name : Bag
Enter Manufacturer Details :
```

```
Name : Amazon
State : Delhi

-----------------------------
Product Serial : 12312
Product Name : Bag
Manufacturer Name : Amazon
State : Delhi
-----------------------------
```