Khoi Duong

Prof. Yang

CS360L

8/3/2022

1.

Source code:

shape.h

```
#ifndef SHAPE_H
#define SHAPE_H

#include <iostream>
using namespace std;

class Shape{
    friend ostream &operator<<( ostream &output, const Shape &s ){
        output << "Shape" << endl;
        return output;
    };
    public:
    double peri, a, size;
    Shape(double s){
        peri = 0;
        a = 0;
        size = s;
    };
    virtual ~Shape(){};
    virtual void setSize(double s){
        size = s;
    };
    virtual void getSize() const{
        cout << "Size of shape: " << size << endl;
    };
    virtual void perimeter() { cout << "Perimeter:" << peri << endl; }
    virtual void area() { cout << "Area:" << a << endl; }
};

#endif // SHAPE_H
```

Circle.h

```
#ifndef CIRCLE_H
#define CIRCLE_H

#include "shape.h"
#include <iostream>
#include <iomanip>
using namespace std;

class Circle: public Shape {
    friend ostream &operator<<( ostream &output, const Circle &c ){
        output << "Circle with radius " << std::setprecision(4) << c.size << ",
perimeter: "
        << std:: setprecision(4) << c.peri << ", and area: " << std::setprecision(4)
<< c.a << endl;
        return output;
    };
    public:
    Circle(double r) : Shape(r) {}
    void setRadius(double r) {
        setSize(r);
    }
    void getRadius() const {
        cout << "Radius of circle: " << std::setprecision(4) << size << endl;
    }
    void perimeter() {
        peri = 2 * 3.14159 * size;
        cout << "Perimeter of circle: " << std::setprecision(4) << peri << endl;
        }
    void area() {
        a = size * size * 3.14159;
        cout << "Area of circle: " << std::setprecision(4) << a << endl;
        }
};

#endif // CIRCLE_H
```

Square.h

```
#ifndef SQUARE_H
#define SQUARE_H

#include "shape.h"
```

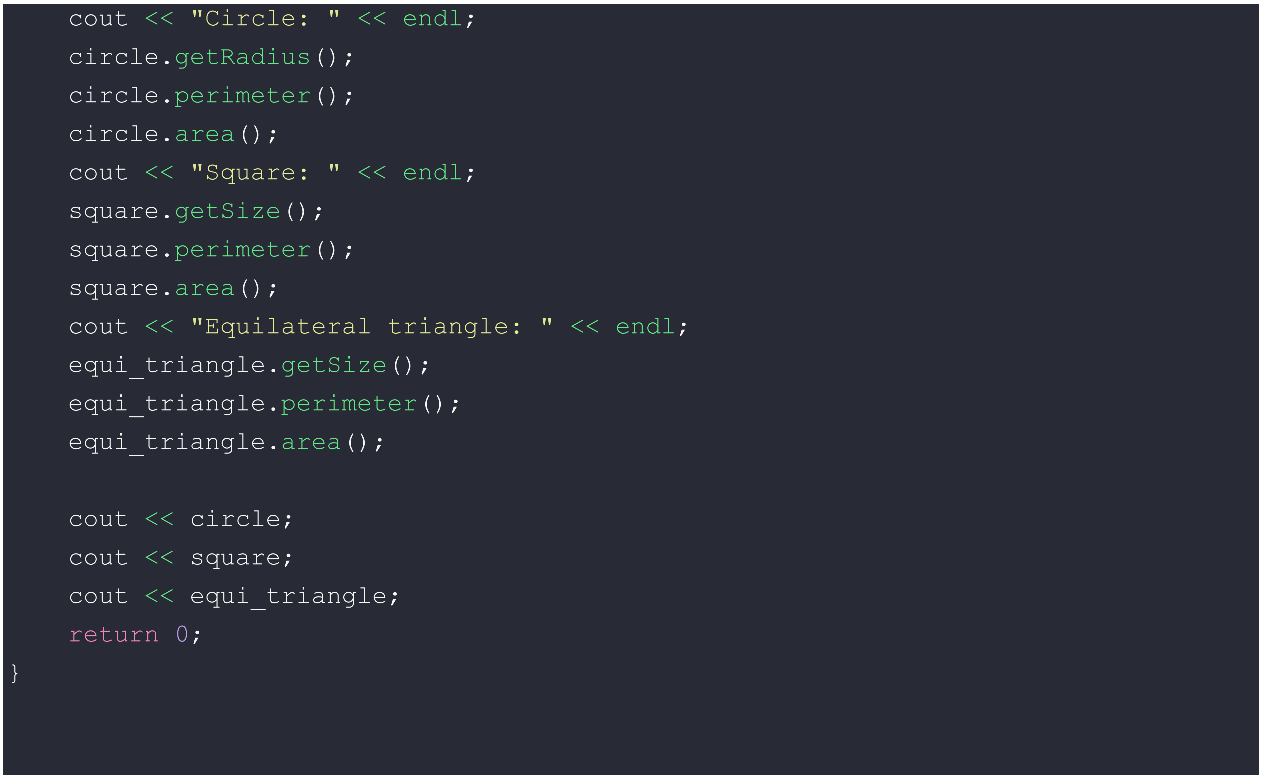
```
#include <iostream>
#include <iomanip>
using namespace std;

class Square: public Shape {
    friend ostream &operator<<( ostream &output, const Square &s ){
        output << "Square with side length " << std::setprecision(4) <<  s.size <<
", perimeter: "
        << std::setprecision(4) << s.peri << ", and area: " << std::setprecision(4)
<< s.a << endl;
        return output;
    };
    public:
    Square(double s) : Shape(s) {}
    void setSize(double s) {
        size = s;
    }
    void getSize() const {
        cout << "Size of square: " << std::setprecision(4) << size << endl;
    }
    void perimeter() {
        peri = 4 * size;
        cout << "Perimeter of square: " << std::setprecision(4) << peri << endl;
        }
    void area() {
        a = size * size;
        cout << "Area of square: " << std::setprecision(4) << a << endl;
        }
};

#endif // CIRCLE_H
```

Equi_triangle.h

```
#ifndef EQUI_TRIANGLE_H
#define EQUI_TRIANGLE_H

#include <cmath>
#include "shape.h"
#include <iostream>
#include <iomanip>
using namespace std;

class Equi_triangle: public Shape {
    friend ostream &operator<<( ostream &output, const Equi_triangle &tri ){
        output << "Equilateral triangle with side length " << std::setprecision(4) << tri.size
        << ", perimeter: " << std::setprecision(4) << tri.peri << ", and area: " << std::setprecision(4) << tri.a << endl;
        return output;
    };
    public:
    Equi_triangle(double s) : Shape(s) {}
    void setSize(double s) {
        size = s;
    }
    void getSize() const {
        cout << "Size of equilateral triangle: " << size << endl;
    }
    void perimeter() {
        peri = 3 * size;
        cout << "Perimeter of equilateral triangle: " << std::setprecision(4) << peri << endl;
        }
    void area() {
        a = 0.5 * size * size;
        cout << "Area of equilateral triangle: " << std::setprecision(4) << a << endl;
        }
};

#endif // CIRCLE_H
```

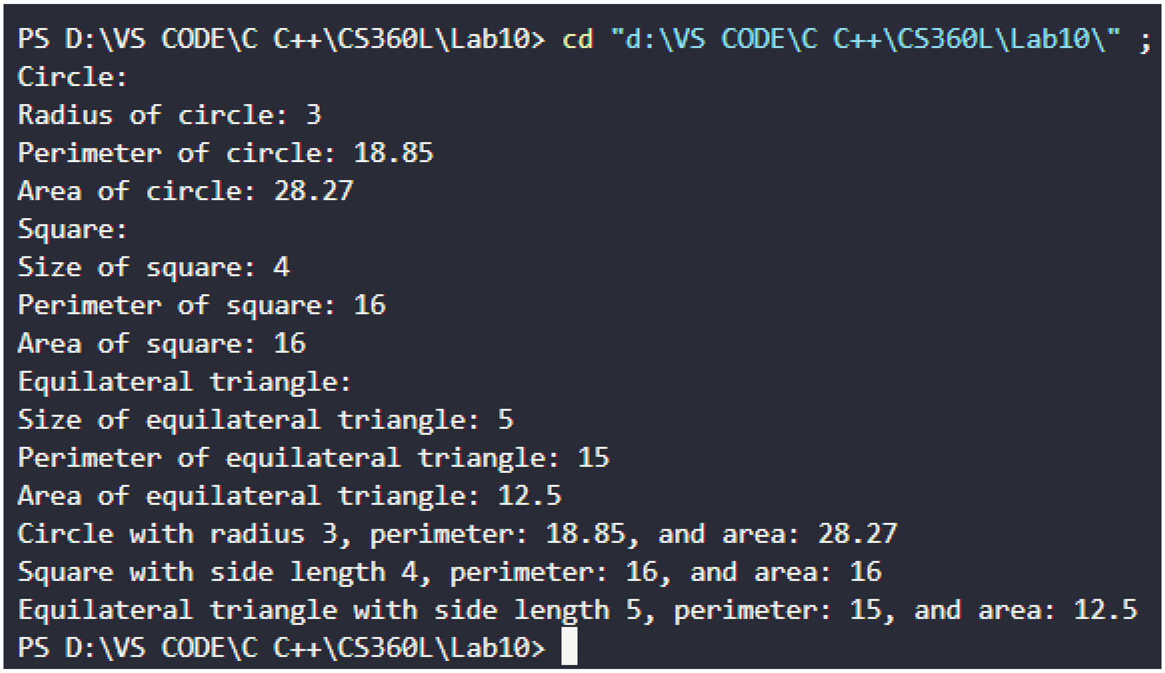
<u>Main.cpp</u>

```
#include <iostream>
#include <iomanip>

using namespace std;
#include "shape.h"
#include "circle.h"
#include "square.h"
#include "equi_triangle.h"

int main(){
    Circle circle(3);
    Square square(4);
    Equi_triangle equi_triangle(5);
```



```cpp
    cout << "Circle: " << endl;
    circle.getRadius();
    circle.perimeter();
    circle.area();
    cout << "Square: " << endl;
    square.getSize();
    square.perimeter();
    square.area();
    cout << "Equilateral triangle: " << endl;
    equi_triangle.getSize();
    equi_triangle.perimeter();
    equi_triangle.area();

    cout << circle;
    cout << square;
    cout << equi_triangle;
    return 0;
}
```

Run program & result:



```
PS D:\VS CODE\C C++\CS360L\Lab10> cd "d:\VS CODE\C C++\CS360L\Lab10\" ;
Circle:
Radius of circle: 3
Perimeter of circle: 18.85
Area of circle: 28.27
Square:
Size of square: 4
Perimeter of square: 16
Area of square: 16
Equilateral triangle:
Size of equilateral triangle: 5
Perimeter of equilateral triangle: 15
Area of equilateral triangle: 12.5
Circle with radius 3, perimeter: 18.85, and area: 28.27
Square with side length 4, perimeter: 16, and area: 16
Equilateral triangle with side length 5, perimeter: 15, and area: 12.5
PS D:\VS CODE\C C++\CS360L\Lab10>
```

2.

Source code:

Person.h

```
#ifndef PERSON_H
#define PERSON_H
#include <iostream>
using namespace std;

class Person{
    friend ostream & operator<< ( ostream & output, const Person & p ){
        output << "Person: " << p.name << " has " << p.head << " head, " << p.eye
<< " eyes, " << p.arm << " arms, and " << p.leg << " legs" << endl;
        return output;
    };
    public:
    string name;
    int age;
    const int eye = 2, arm = 2, head = 1, leg = 2;
    Person(string n, int a){
        name = n;
        age = a;
    };
    ~Person(){};
    void getName() const {
        cout << "Name of person: " << name << endl;
    }
    void getAge() const {
        cout << "Age of person: " << age << endl;
    }
    void setName(string n){
        name = n;
    }
    void setAge(int a){
        age = a;
    }
    virtual void Think() const {
        cout << "Person is thinking using either left or right brain, or both." <<
endl;
    }
    virtual void redbloodCells() const {
        cout << "Person has an average of 4.7 to 5.4 million cells per microliter
of blood." << endl;
```

```
    }
    virtual void bloodPressure() const{
        cout << "Blood pressure: average" << endl;
    }
    virtual void mainHormone() const{
        cout << "Hormone: estrogen, progesterone, testosterone" << endl;
    }
    virtual void Reproductive() const {
        cout << "Reproductive: different based on gender" << endl;
    }
};

#endif // PERSON_H
```

Man.h

```
#ifndef MAN_H
#define MAN_H
#include <iostream>
using namespace std;
#include "person.h"

class Man : public Person {
    friend ostream & operator<< ( ostream & output, const Man & man ){
        output << "Man: " << man.name << " has a gen code of " << man.gen << ", "
<< man.head << " head, " << man.eye << " eyes, " << man.arm << " arms, and " <<
man.leg << " legs" << endl;
        return output;
    };
    string gen;
    public:
    Man(string n, int a) : Person(n, a) {
        gen = "XY";
    }
    ~Man() {}
    void Think(){
        cout << "Men mostly think with their left brain. " << endl;
        cout << "Men think more with grey matter. " << endl;
    }
    void redbloodCells() const {
        cout << "Person has an average of 4.7 to 6.1 million cells per microliter
of blood." << endl;
    }
    void bloodPressure() const {
        cout << "Blood pressure: higher" << endl;
    }
    virtual void mainHormone() const{
        cout << "Hormone (main): testosterone" << endl;
    }
    virtual void Reproductive() const {
        cout << "Reproductive: testes and sperm" << endl;
    }
};

#endif //MAN_H
```

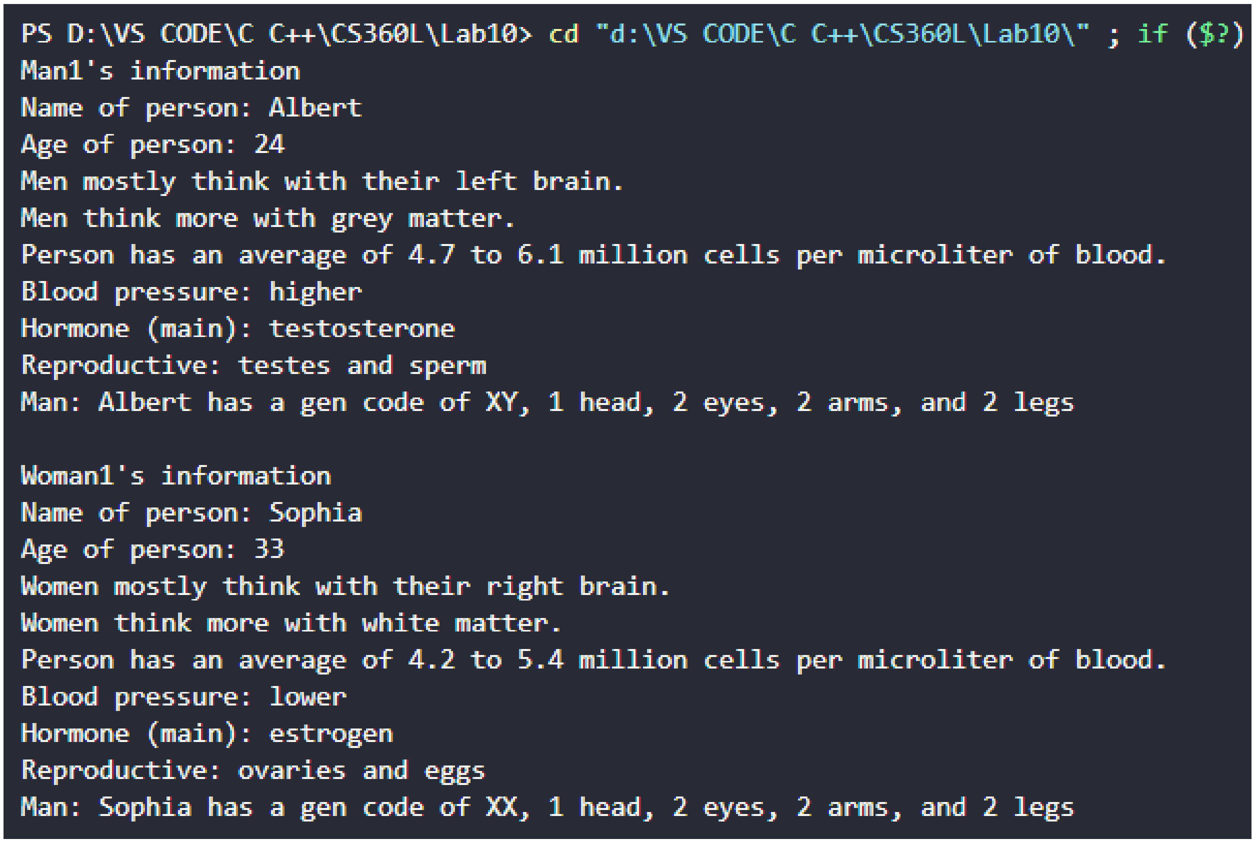
Woman.h

```
#ifndef WOMAN_H
#define WOMAN_H
#include <iostream>
using namespace std;
#include "person.h"

class Woman : public Person {
    friend ostream & operator<< ( ostream & output, const Woman & woman ){
        output << "Man: " << woman.name << " has a gen code of " << woman.gen << ", " << woman.head << " head, " << woman.eye << " eyes, " << woman.arm << " arms, and " << woman.leg << " legs" << endl;
        return output;
    };
    string gen;
    public:
    Woman(string n, int a) : Person(n, a) {
        gen = "XX";
    }
    ~Woman() {}
    void Think() {
        cout << "Women mostly think with their right brain." << endl;
        cout << "Women think more with white matter." << endl;
    }
    void redbloodCells() const {
        cout << "Person has an average of 4.2 to 5.4 million cells per microliter of blood." << endl;
    }
    void bloodPressure() const {
        cout << "Blood pressure: lower" << endl;
    }
    virtual void mainHormone() const{
        cout << "Hormone (main): estrogen" << endl;
    }
    virtual void Reproductive() const {
        cout << "Reproductive: ovaries and eggs" << endl;
    }
};

#endif //WOMAN_H
```

Main.cpp

```
#include <iostream>
#include "person.h"
#include "man.h"
#include "woman.h"
using namespace std;

int main(){
    Man man1("Albert", 24);
    Woman woman1("Sophia", 33);
    cout << "Man1's information" << endl;
    man1.getName();
    man1.getAge();
    man1.Think();
    man1.redbloodCells();
    man1.bloodPressure();
    man1.mainHormone();
    man1.Reproductive();
    cout << man1 << endl;

    cout << "Woman1's information" << endl;
    woman1.getName();
    woman1.getAge();
    woman1.Think();
    woman1.redbloodCells();
    woman1.bloodPressure();
    woman1.mainHormone();
    woman1.Reproductive();
    cout << woman1 << endl;
    return 0;
}
```

Run program & result:

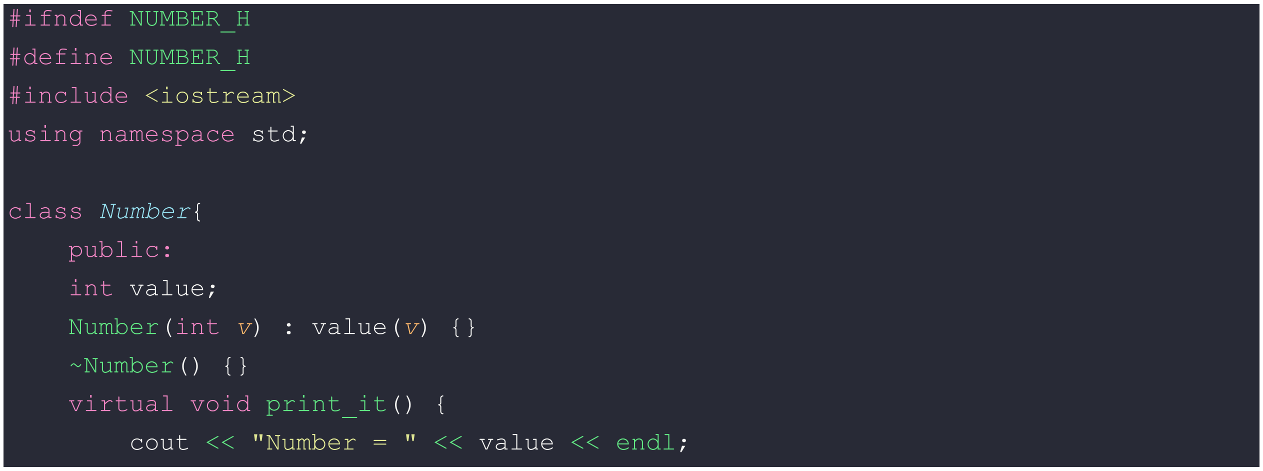


```
PS D:\VS CODE\C C++\CS360L\Lab10> cd "d:\VS CODE\C C++\CS360L\Lab10\" ; if ($?)
Man1's information
Name of person: Albert
Age of person: 24
Men mostly think with their left brain.
Men think more with grey matter.
Person has an average of 4.7 to 6.1 million cells per microliter of blood.
Blood pressure: higher
Hormone (main): testosterone
Reproductive: testes and sperm
Man: Albert has a gen code of XY, 1 head, 2 eyes, 2 arms, and 2 legs

Woman1's information
Name of person: Sophia
Age of person: 33
Women mostly think with their right brain.
Women think more with white matter.
Person has an average of 4.2 to 5.4 million cells per microliter of blood.
Blood pressure: lower
Hormone (main): estrogen
Reproductive: ovaries and eggs
Man: Sophia has a gen code of XX, 1 head, 2 eyes, 2 arms, and 2 legs
```

3.

Source code:

Number.h



```
#ifndef NUMBER_H
#define NUMBER_H
#include <iostream>
using namespace std;

class Number{
    public:
    int value;
    Number(int v) : value(v) {}
    ~Number() {}
    virtual void print_it() {
        cout << "Number = " << value << endl;
    }
};

#endif //NUMBER_H
```

Binary.h

```
#ifndef BINARY_H
#define BINARY_H
#include <iostream>
#include "number.h"
using namespace std;

class Binary : public Number {
    public:
    Binary(int v) : Number(v) {}
    ~Binary() {}
    void dectoBinary(int n){
        int bin[32];
        int i = 0;
        while (n>0) {
            bin[i] = n%2;
            n /= 2;
            i++;
        }
        for (int j = i-1; j >= 0; j--) {
            cout << bin[j];
        }
    }
    void print_it() {
        cout << "Binary = ";
        dectoBinary(value);
        cout << endl;
    }
};

#endif //Binary_H
```

Hex.h

```
#ifndef HEX_H
#define HEX_H
#include <iostream>
```

```
#include <sstream>
#include "number.h"
using namespace std;

class Hex : public Number {
    public:
    Hex(int v) : Number(v) {}
    ~Hex() {}
        void print_it() {
            std::ostringstream ss;
            ss << std::hex << value;
            string s = ss.str();
            cout << "Hex = " << s << endl;
    }
};

#endif //HEX_H
```

Octal.h

```
#ifndef OCTAL_H
#define OCTAL_H
#include <iostream>
#include "number.h"
using namespace std;

class Octal : public Number {
    public:
    Octal(int v) : Number(v) {}
    ~Octal() {}
    int integertoOctal(int n){
        int remainder;
        int octal = 0, i = 1;
        while (n != 0)
        {
            remainder = n % 8;
            n /= 8;
            octal += remainder * i;
            i *= 10;
        }
        return octal;
    }
    void print_it() {
        int s = integertoOctal(value);
        cout << "Octal = " << s << endl;
    }
};

#endif //Octal_H
```

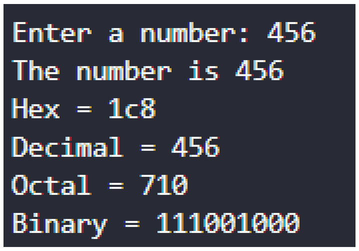
Decimal.h

```
#ifndef DECIMAL_H
#define DECIMAL_H
#include <iostream>
#include "number.h"
using namespace std;

class Decimal : public Number {
    public:
    Decimal(int v) : Number(v) {}
    ~Decimal() {}
    void print_it() {
        cout << "Decimal = " << dec << value << endl;
    }
};

#endif //Decimal_H
```

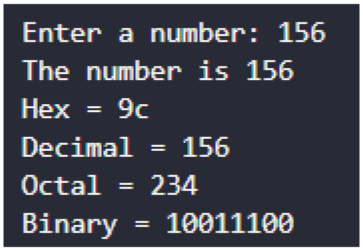
Main.cpp

```
#include <iostream>
#include <typeinfo>
#include "number.h"
#include "hex.h"
#include "decimal.h"
#include "octal.h"
#include "binary.h"
using namespace std;

int main(){
    int a;
    cout << "Enter a number: ";
    cin >> a;
    cout << "The number is " << a << endl;
    Hex h(a);
    h.print_it();
    Decimal d(a);
    d.print_it();
    Octal o(a);
    o.print_it();
    Binary b(a);
    b.print_it();
    return 0;
}
```

Run program & result:



```
Enter a number: 456
The number is 456
Hex = 1c8
Decimal = 456
Octal = 710
Binary = 111001000
```



```
Enter a number: 156
The number is 156
Hex = 9c
Decimal = 156
Octal = 234
Binary = 10011100
```