UNIVERSITY OF ENGINEERING AND TECHNOLOGY

(NAROWAL CAMPUS)

Object-Oriented Programming Lab Manual

Created by: Muhammad Abdullah

Registration Number: 2022-CS-525

Topics: Inheritance Default and Parameterized Constructor,

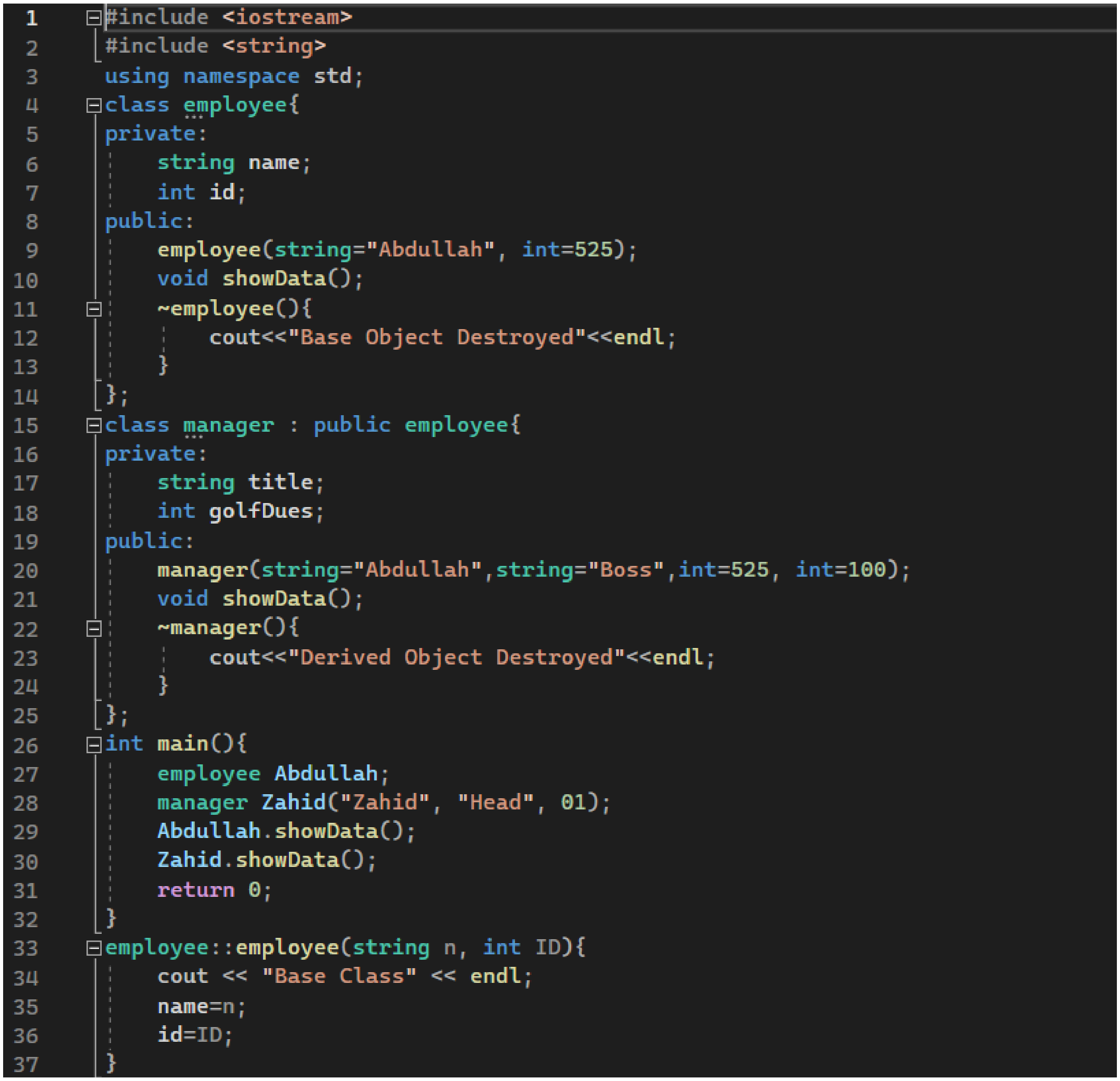
Inheritance Function Overriding, Virtual Functions,

Static and Dynamic Binding & Pure Virtual Function

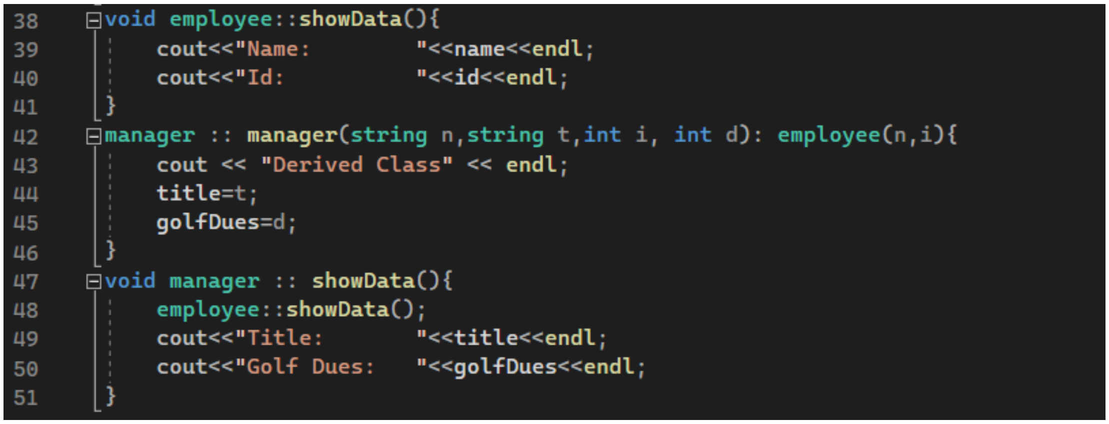
- ## Task 1:

Write a program of inheritance in C++ in which the inheritance of classes occur and demonstrate the call of base class constructors.

Program:

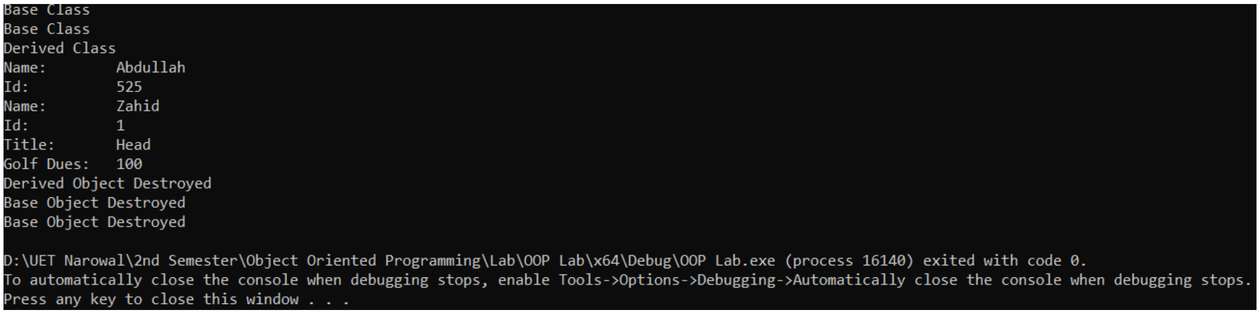


```
#include <iostream>
#include <string>
using namespace std;
class employee{
private:
    string name;
    int id;
public:
    employee(string="Abdullah", int=525);
    void showData();
    ~employee(){
        cout<<"Base Object Destroyed"<<endl;
    }
};
class manager : public employee{
private:
    string title;
    int golfDues;
public:
    manager(string="Abdullah",string="Boss",int=525, int=100);
    void showData();
    ~manager(){
        cout<<"Derived Object Destroyed"<<endl;
    }
};
int main(){
    employee Abdullah;
    manager Zahid("Zahid", "Head", 01);
    Abdullah.showData();
    Zahid.showData();
    return 0;
}
employee::employee(string n, int ID){
    cout << "Base Class" << endl;
    name=n;
    id=ID;
}
```



```cpp
void employee::showData(){
    cout<<"Name:        "<<name<<endl;
    cout<<"Id:          "<<id<<endl;
}
manager :: manager(string n,string t,int i, int d): employee(n,i){
    cout << "Derived Class" << endl;
    title=t;
    golfDues=d;
}
void manager :: showData(){
    employee::showData();
    cout<<"Title:       "<<title<<endl;
    cout<<"Golf Dues:   "<<golfDues<<endl;
}
```

Output:



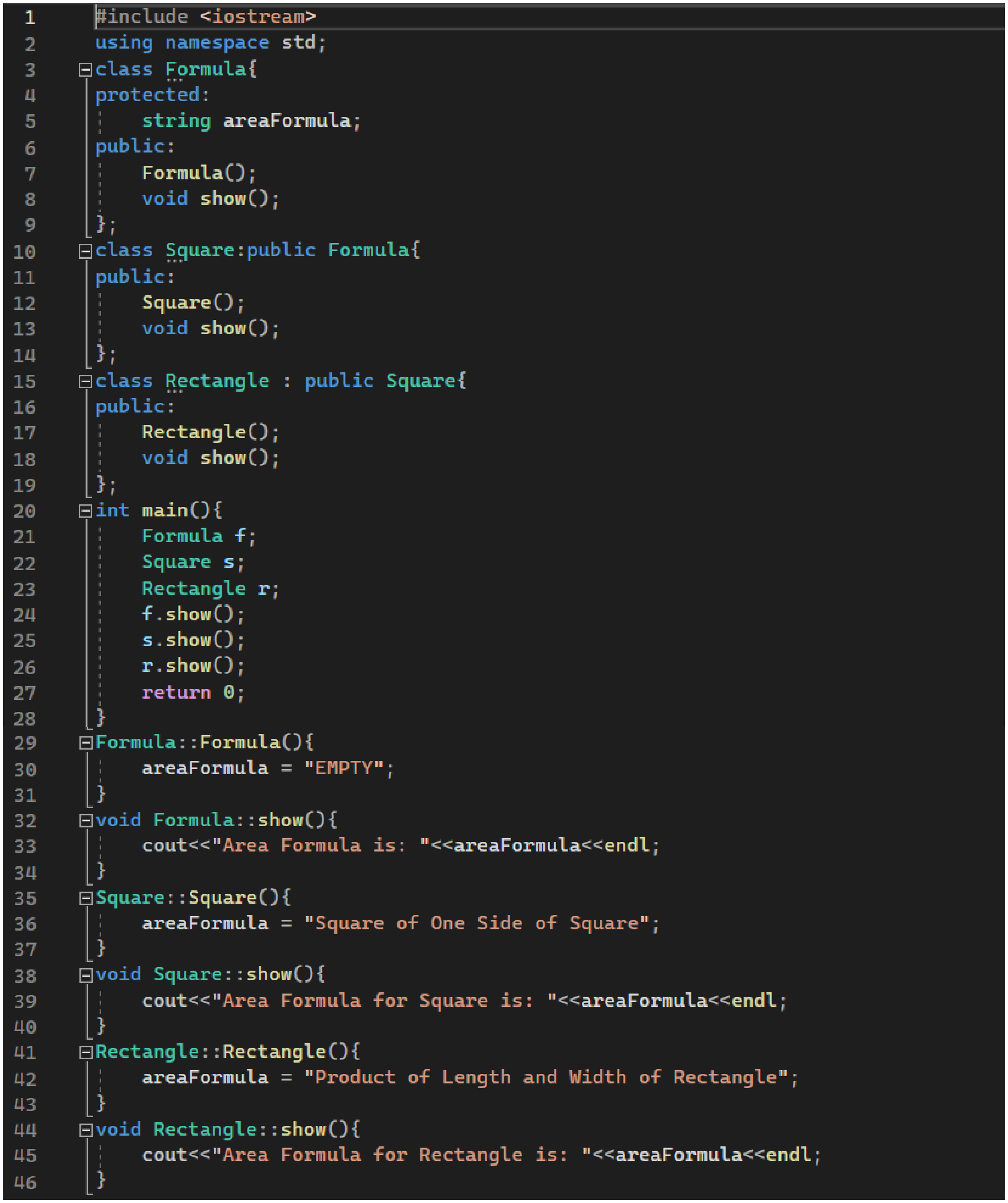
```
Base Class
Base Class
Derived Class
Name:        Abdullah
Id:          525
Name:        Zahid
Id:          1
Title:       Head
Golf Dues:   100
Derived Object Destroyed
Base Object Destroyed
Base Object Destroyed

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 16140) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

- ## Task 2:

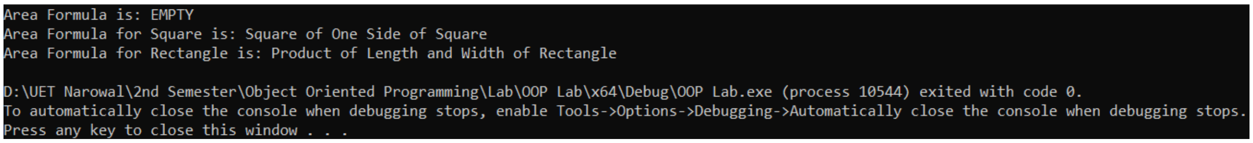
Override the member functions of a base class which is inherited in a C++ program.

Program:



```cpp
#include <iostream>
using namespace std;
class Formula{
protected:
    string areaFormula;
public:
    Formula();
    void show();
};
class Square:public Formula{
public:
    Square();
    void show();
};
class Rectangle : public Square{
public:
    Rectangle();
    void show();
};
int main(){
    Formula f;
    Square s;
    Rectangle r;
    f.show();
    s.show();
    r.show();
    return 0;
}
Formula::Formula(){
    areaFormula = "EMPTY";
}
void Formula::show(){
    cout<<"Area Formula is: "<<areaFormula<<endl;
}
Square::Square(){
    areaFormula = "Square of One Side of Square";
}
void Square::show(){
    cout<<"Area Formula for Square is: "<<areaFormula<<endl;
}
Rectangle::Rectangle(){
    areaFormula = "Product of Length and Width of Rectangle";
}
void Rectangle::show(){
    cout<<"Area Formula for Rectangle is: "<<areaFormula<<endl;
}
```

Output:



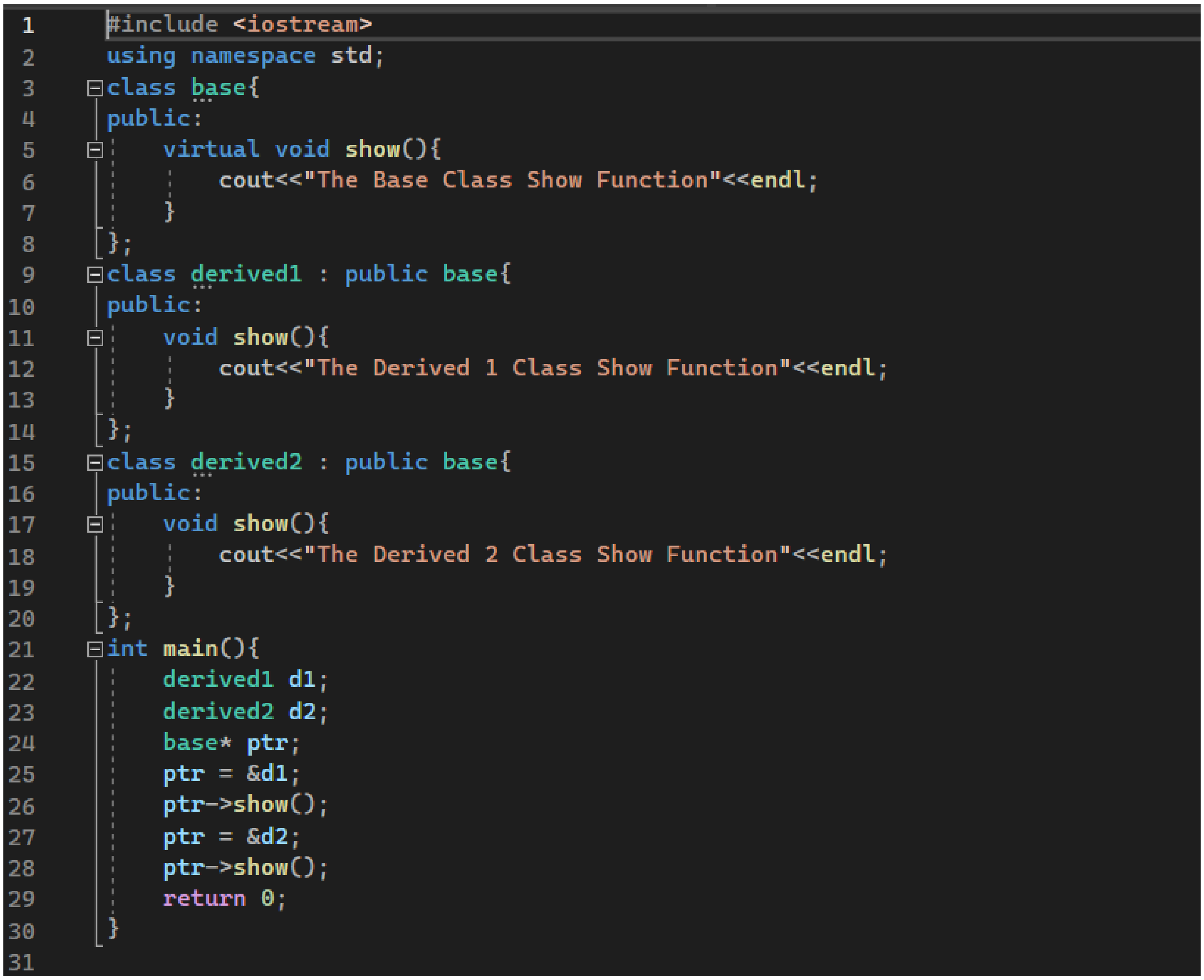
```
Area Formula is: EMPTY
Area Formula for Square is: Square of One Side of Square
Area Formula for Rectangle is: Product of Length and Width of Rectangle

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 10544) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

- ## Task 3:

Write a C++ program in which a base class contains a virtual function.

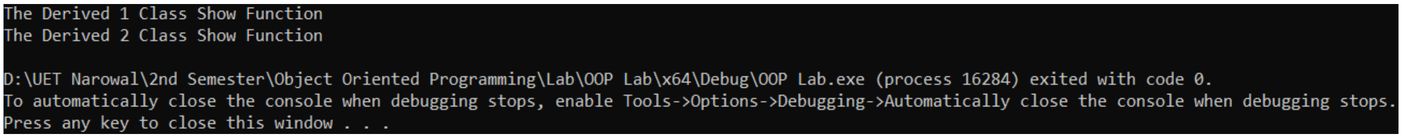
Program:



```cpp
#include <iostream>
using namespace std;
class base{
public:
    virtual void show(){
        cout<<"The Base Class Show Function"<<endl;
    }
};
class derived1 : public base{
public:
    void show(){
        cout<<"The Derived 1 Class Show Function"<<endl;
    }
};
class derived2 : public base{
public:
    void show(){
        cout<<"The Derived 2 Class Show Function"<<endl;
    }
};
int main(){
    derived1 d1;
    derived2 d2;
    base* ptr;
    ptr = &d1;
    ptr->show();
    ptr = &d2;
    ptr->show();
    return 0;
}

```

Output:



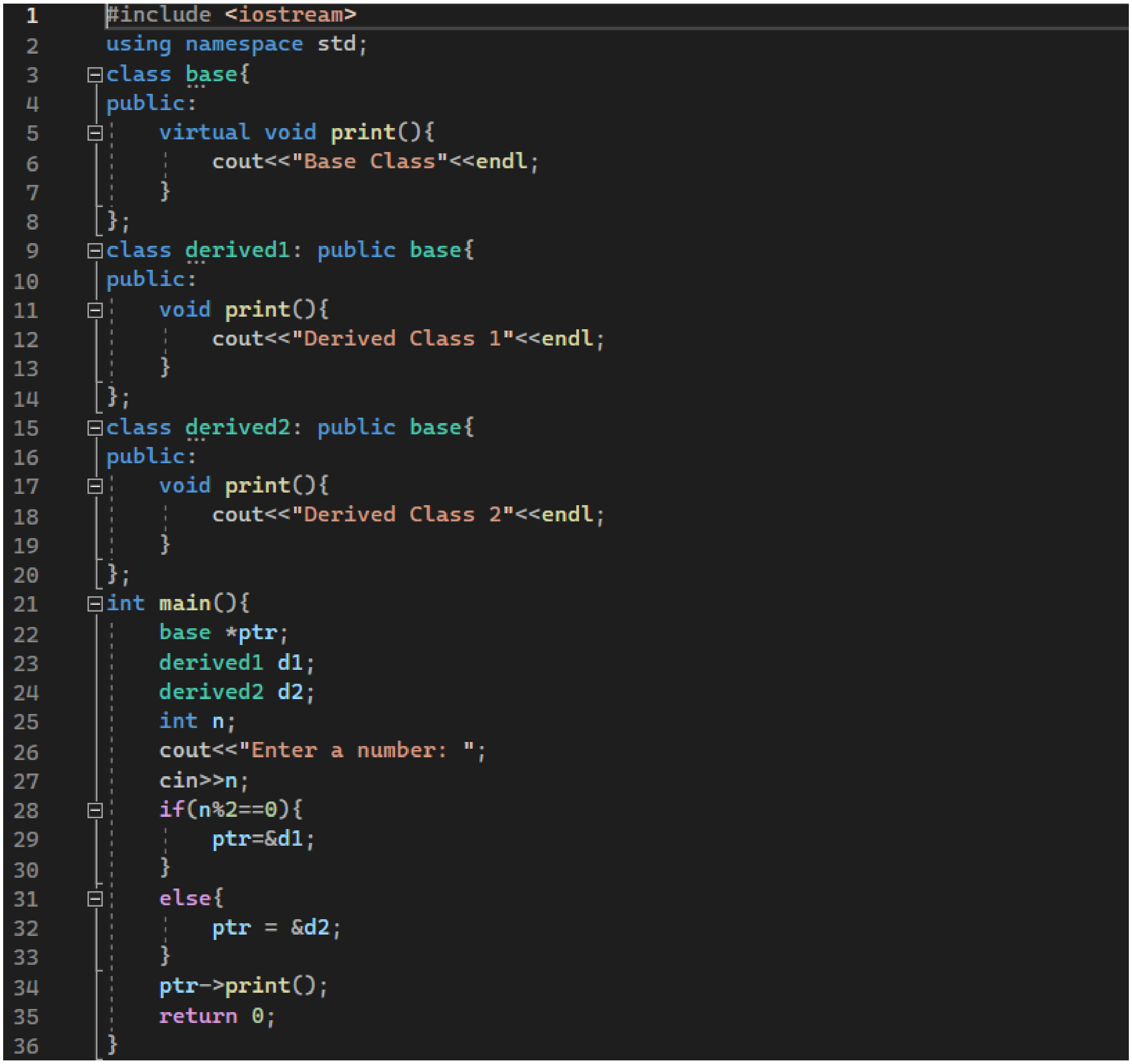
```
The Derived 1 Class Show Function
The Derived 2 Class Show Function

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 16284) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

- ## **Task 4:**

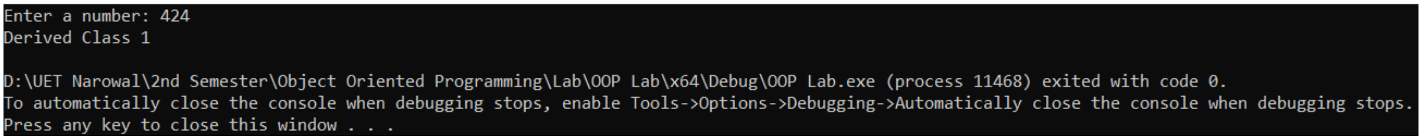
Write a program of dynamic binding in C++.

Program:



```cpp
#include <iostream>
using namespace std;
class base{
public:
    virtual void print(){
        cout<<"Base Class"<<endl;
    }
};
class derived1: public base{
public:
    void print(){
        cout<<"Derived Class 1"<<endl;
    }
};
class derived2: public base{
public:
    void print(){
        cout<<"Derived Class 2"<<endl;
    }
};
int main(){
    base *ptr;
    derived1 d1;
    derived2 d2;
    int n;
    cout<<"Enter a number: ";
    cin>>n;
    if(n%2==0){
        ptr=&d1;
    }
    else{
        ptr = &d2;
    }
    ptr->print();
    return 0;
}
```

Output:



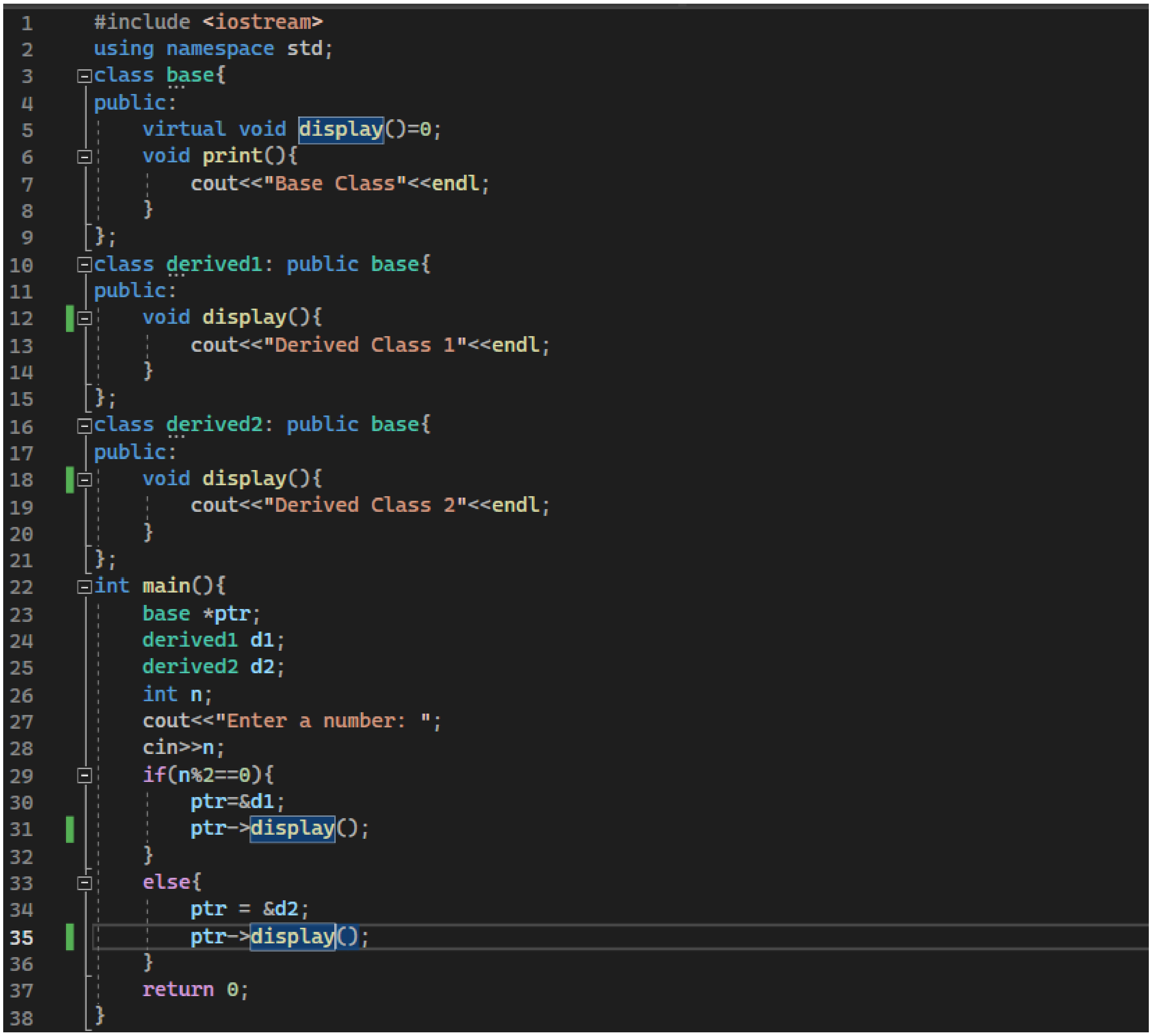
```
Enter a number: 424
Derived Class 1

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 11468) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

- ## **Task 5:**

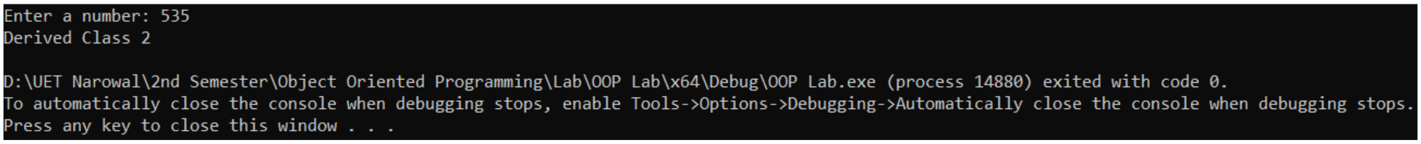
Create a class that acts as an abstract class and contains a pure virtual function.

Program:



```cpp
#include <iostream>
using namespace std;
class base{
public:
    virtual void display()=0;
    void print(){
        cout<<"Base Class"<<endl;
    }
};
class derived1: public base{
public:
    void display(){
        cout<<"Derived Class 1"<<endl;
    }
};
class derived2: public base{
public:
    void display(){
        cout<<"Derived Class 2"<<endl;
    }
};
int main(){
    base *ptr;
    derived1 d1;
    derived2 d2;
    int n;
    cout<<"Enter a number: ";
    cin>>n;
    if(n%2==0){
        ptr=&d1;
        ptr->display();
    }
    else{
        ptr = &d2;
        ptr->display();
    }
    return 0;
}
```

Output:



```
Enter a number: 535
Derived Class 2

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 14880) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```
