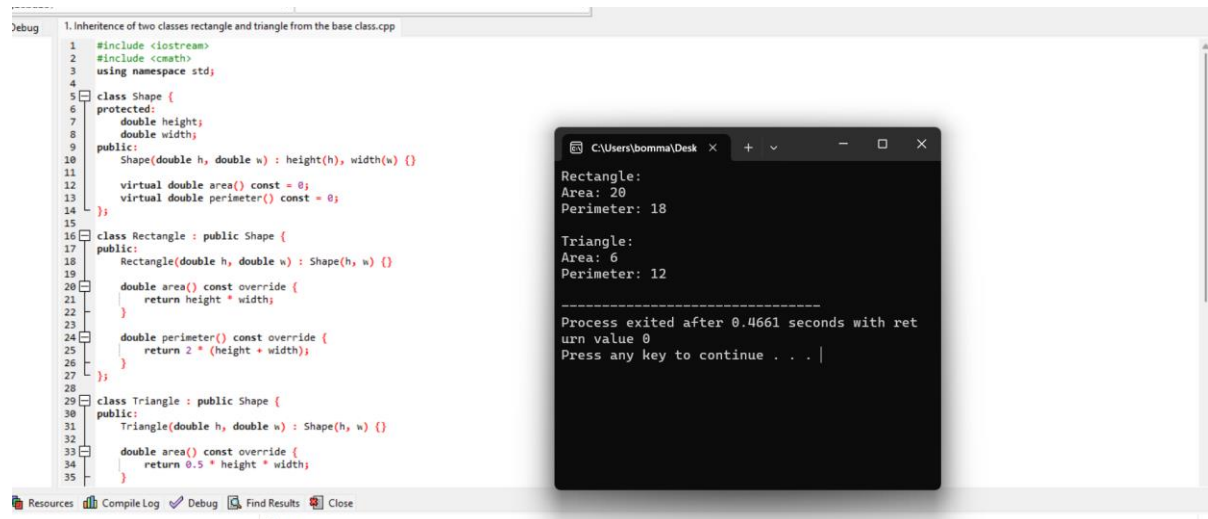


1. Create a base class called Shape with data members for height and width. Derive two classes Rectangle and Triangle from the base class. Write member functions to calculate the area and perimeter of each class



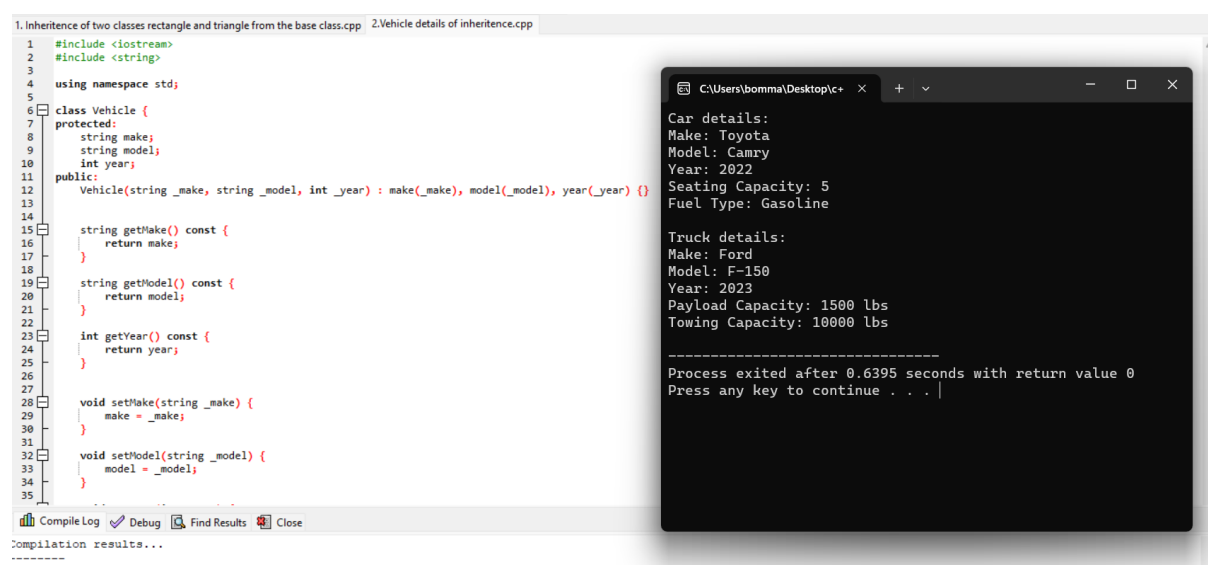
```
1. Inheritance of two classes rectangle and triangle from the base class.cpp
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  class Shape {
6  protected:
7      double height;
8      double width;
9  public:
10     Shape(double h, double w) : height(h), width(w) {}
11
12     virtual double area() const = 0;
13     virtual double perimeter() const = 0;
14 };
15
16 class Rectangle : public Shape {
17 public:
18     Rectangle(double h, double w) : Shape(h, w) {}
19
20     double area() const override {
21         return height * width;
22     }
23
24     double perimeter() const override {
25         return 2 * (height + width);
26     }
27 };
28
29 class Triangle : public Shape {
30 public:
31     Triangle(double h, double w) : Shape(h, w) {}
32
33     double area() const override {
34         return 0.5 * height * width;
35     }
36 }
```

```
C:\Users\bomma\Desktop
Rectangle:
Area: 20
Perimeter: 18

Triangle:
Area: 6
Perimeter: 12

-----
Process exited after 0.4661 seconds with return value 0
Press any key to continue . . . |
```

2. Create a base class called vehicle with data members for make, model, and year. Derive two classes Car and Truck from the base class. The Car class should have additional data members for seating capacity and fuel type, while the Truck class should have additional data members for payload capacity and towing capacity. Write member functions to get and set the data members for each class



```
1. Inheritance of two classes rectangle and triangle from the base class.cpp 2. Vehicle details of inheritance.cpp
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class Vehicle {
7  protected:
8      string make;
9      string model;
10     int year;
11 public:
12     Vehicle(string _make, string _model, int _year) : make(_make), model(_model), year(_year) {}
13
14
15     string getMake() const {
16         return make;
17     }
18
19     string getModel() const {
20         return model;
21     }
22
23     int getYear() const {
24         return year;
25     }
26
27
28     void setMake(string _make) {
29         make = _make;
30     }
31
32     void setModel(string _model) {
33         model = _model;
34     }
35 }
```

```
C:\Users\bomma\Desktop\c++
Car details:
Make: Toyota
Model: Camry
Year: 2022
Seating Capacity: 5
Fuel Type: Gasoline

Truck details:
Make: Ford
Model: F-150
Year: 2023
Payload Capacity: 1500 lbs
Towing Capacity: 10000 lbs

-----
Process exited after 0.6395 seconds with return value 0
Press any key to continue . . . |
```

3. Create a base class called Animal with data members for name, species, and age. Derive two classes Cat and Dog from the base class. The Cat class should have additional data members for color and breed, while the Dog class should have additional data members for weight and breed. Write member functions to get and set the data members for each class

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class Animal {
5 protected:
6     string name;
7     string species;
8     int age;
9 public:
10     Animal(const string& _name, const string& _species, int _age)
11         : name(_name), species(_species), age(_age) {}
12     string getName() const { return name; }
13     void setName(const string& _name) { name = _name; }
14     string getSpecies() const { return species; }
15     void setSpecies(const string& _species) { species = _species; }
16     int getAge() const { return age; }
17     void setAge(int _age) { age = _age; }
18 };
19
20 class Cat : public Animal {
21 private:
22     string color;
23     string breed;
24 public:
25     Cat(const string& _name, const string& _species, int _age, const string& _color, const string& _breed)
26         : Animal(_name, _species, _age), color(_color), breed(_breed) {}
27     string getColor() const { return color; }
28     void setColor(const string& _color) { color = _color; }
29 };
30
31 class Dog : public Animal {
32 private:
33     double weight;
34     string breed;
35 public:
36     Dog(const string& _name, const string& _species, int _age, double _weight, const string& _breed)
37         : Animal(_name, _species, _age), weight(_weight), breed(_breed) {}
38     double getWeight() const { return weight; }
39     void setWeight(double _weight) { weight = _weight; }
40     string getBreed() const { return breed; }
41     void setBreed(const string& _breed) { breed = _breed; }
42 };
43
44 int main() {
45     Cat cat("Fluffy", "Cat", 5, "White", "Persian");
46     Dog dog("Buddy", "Dog", 3, 25.5, "Golden Retriever");
47     cat.print();
48     dog.print();
49     return 0;
50 }

```

```

Cat:
Name: Fluffy
Species: Cat
Age: 5
Color: White
Breed: Persian

Dog:
Name: Buddy
Species: Dog
Age: 3
Weight: 25.5
Breed: Golden Retriever

-----
Process exited after 0.7047 seconds with return value 0
Press any key to continue . . .

```

4. Create a base class called Employee with data members for name, id, and salary. Derive two classes Manager and Engineer from the base class. The Manager class should have additional data members for department and bonus, while the Engineer class should have additional data members for specialty and hours. Write member functions to get and set the data members for each class

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 class Employee {
6 protected:
7     string name;
8     int id;
9     double salary;
10 public:
11     Employee(const string& n, int i, double s) : name(n), id(i), salary(s) {}
12     void setName(const string& n) { name = n; }
13     string getName() const { return name; }
14     void setId(int i) { id = i; }
15     int getId() const { return id; }
16     void setSalary(double s) { salary = s; }
17     double getSalary() const { return salary; }
18 };
19
20 class Manager : public Employee {
21 private:
22     string department;
23     double bonus;
24 public:
25     Manager(const string& n, int i, double s, const string& dept, double b)
26         : Employee(n, i, s), department(dept), bonus(b) {}
27     void setDepartment(const string& dept) { department = dept; }
28     string getDepartment() const { return department; }
29     void setBonus(double b) { bonus = b; }
30     double getBonus() const { return bonus; }
31 };
32
33 class Engineer : public Employee {
34 private:
35     string specialty;
36     int hours;
37 public:
38     Engineer(const string& n, int i, double s, const string& spec, int h)
39         : Employee(n, i, s), specialty(spec), hours(h) {}
40     void setSpecialty(const string& spec) { specialty = spec; }
41     string getSpecialty() const { return specialty; }
42     void setHours(int h) { hours = h; }
43     int getHours() const { return hours; }
44 };
45
46 int main() {
47     Manager manager("John Doe", 101, 60000, "Marketing", 5000);
48     Engineer engineer("Alice Smith", 201, 70000, "Software Development", 40);
49     manager.print();
50     engineer.print();
51     return 0;
52 }

```

```

Manager:
Name: John Doe
ID: 101
Salary: $60000
Department: Marketing
Bonus: $5000

Engineer:
Name: Alice Smith
ID: 201
Salary: $70000
Specialty: Software Development
Hours: 40

-----
Process exited after 0.7057 seconds with return value 0
Press any key to continue . . .

```

5. Create a base class called Person with data members for name, age, and gender. Derive two classes Student and Teacher from the base class. The Student class should have additional data members for roll number and class, while the Teacher class should have additional data members for subject and salary. Write member functions to get and set the data members for each class.

```

// Person.h
class Person {
public:
    string name;
    int age;
    char gender;

    Person(const string& n, int a, char g) : name(n), age(a), gender(g) {}

    void setName(const string& n) { name = n; }
    string getName() const { return name; }

    void setAge(int a) { age = a; }
    int getAge() const { return age; }

    void setGender(char g) { gender = g; }
    char getGender() const { return gender; }
};

// Student.h
class Student : public Person {
private:
    int rollNumber;
    string className;
public:
    Student(const string& n, int a, char g, int roll, const string& cls)
        : Person(n, a, g), rollNumber(roll), className(cls) {}

    void setRollNumber(int roll) { rollNumber = roll; }
    int getRollNumber() const { return rollNumber; }

    void setClassName(const string& cls) { className = cls; }
    string getClassName() const { return className; }
};

// Teacher.h
class Teacher : public Person {
private:
    string subject;
    double salary;
public:
    Teacher(const string& n, int a, char g, const string& sub, double sal)
        : Person(n, a, g), subject(sub), salary(sal) {}

    void setSubject(const string& sub) { subject = sub; }
    string getSubject() const { return subject; }

    void setSalary(double sal) { salary = sal; }
    double getSalary() const { return salary; }
};
    
```

```

// main.cpp
int main() {
    Student s("Prashanth", 20, 'M', 192110624, "Physics");
    s.setName("prashanth");
    s.setAge(20);
    s.setGender('M');
    s.setRollNumber(192110624);
    s.setClassName("Physics");

    Teacher t("Raju", 35, 'M', "Mathematics", 60000);
    t.setName("Raju");
    t.setAge(35);
    t.setGender('M');
    t.setSubject("Mathematics");
    t.setSalary(60000);

    return 0;
}
    
```

```

Student:
Name: prashanth
Age: 20
Gender: M
Roll Number: 192110624
Class: Physics

Teacher:
Name: Raju
Age: 35
Gender: M
Subject: Mathematics
Salary: $60000

Process exited after 0.5754 seconds with return value 0
Press any key to continue . . .
    
```

POINTERS PROGRAMS:

6. Write a C++ program to create a pointer to an integer and display its value.

```

// main.cpp
#include <iostream>
using namespace std;

int main() {
    int num = 10;
    int *ptr = &num;

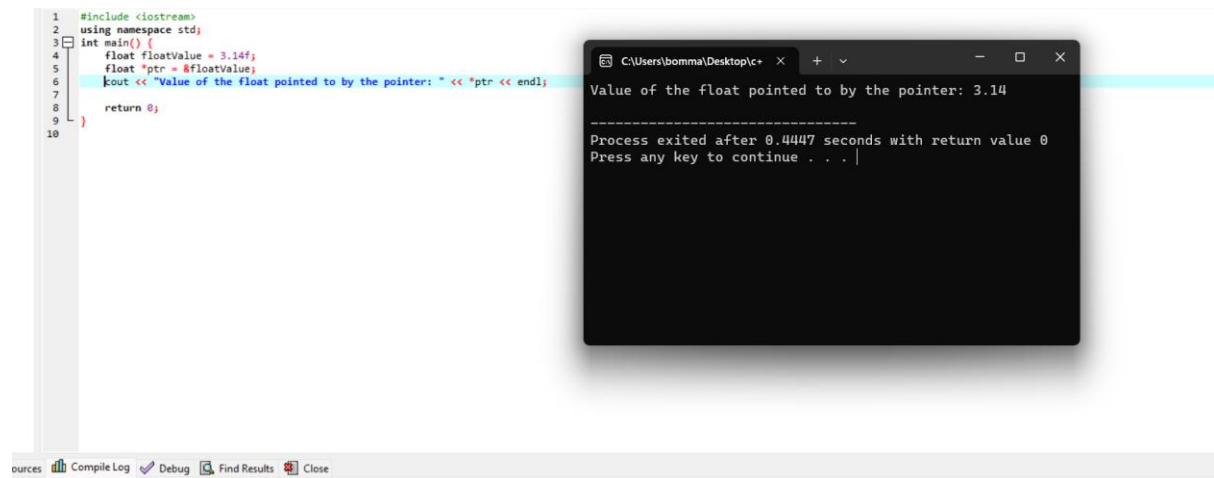
    cout << "Value of the integer pointed to by the pointer: " << *ptr << endl;

    return 0;
}
    
```

```

Process exited after 0.459 seconds with return value 0
Press any key to continue . . .
    
```

7. Write a C++ program to create a pointer to a float and display its value.



The screenshot shows a C++ IDE with a source code editor on the left and a console window on the right. The source code is as follows:

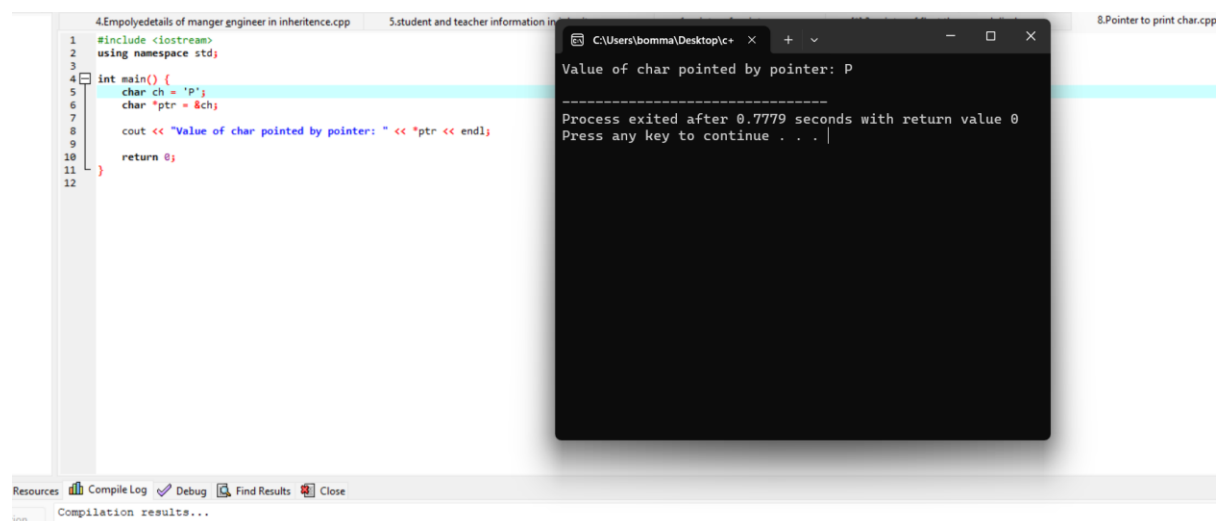
```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     float floatValue = 3.14f;
5     float *ptr = &floatValue;
6     cout << "Value of the float pointed to by the pointer: " << *ptr << endl;
7 }
8 return 0;
9
10
```

The console window displays the output of the program:

```
C:\Users\bomma\Desktop\c++ x + -
Value of the float pointed to by the pointer: 3.14

-----
Process exited after 0.4447 seconds with return value 0
Press any key to continue . . .
```

8. Write a C++ program to create a pointer to a char and display its value.



The screenshot shows a C++ IDE with a source code editor on the left and a console window on the right. The source code is as follows:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     char ch = 'P';
6     char *ptr = &ch;
7     cout << "Value of char pointed by pointer: " << *ptr << endl;
8 }
9 return 0;
10
11
12
```

The console window displays the output of the program:

```
C:\Users\bomma\Desktop\c++ x + -
Value of char pointed by pointer: P

-----
Process exited after 0.7779 seconds with return value 0
Press any key to continue . . .
```

9. Write a C++ program to create a pointer to a double and display its value.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     double num = 3.14159;
6     double *ptr = &num;
7     cout << "Value of double pointed by pointer: " << *ptr << endl;
8
9     return 0;
10 }
```

Value of double pointed by pointer: 3.14159

Process exited after 0.6592 seconds with return value 0
Press any key to continue . . .

10. Write a C++ program to create a pointer to a string and display its value.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main() {
6     string str = "Hello, World!";
7     string *ptr = &str;
8     cout << "Value of string pointed by pointer: " << *ptr << endl;
9
10    return 0;
11 }
```

Value of string pointed by pointer: Hello, World!

Process exited after 0.5977 seconds with return value 0
Press any key to continue . . .

11. Write a C++ program to create a pointer to an array of elements and display its value.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int arr[] = {1, 2, 3, 4, 5};
6     int *ptr = arr;
7
8     cout << "Values of array elements pointed by pointer: ";
9     for (int i = 0; i < 5; ++i) {
10         cout << *(ptr + i) << " ";
11     }
12     cout << endl;
13
14     return 0;
15 }
```

```
-----
Process exited after 0.8768 seconds with return value
0
Press any key to continue . . . |
```

Compilation results...

- Errors: 0

- Warnings: 0

- Output Filename: C:\Users\bomma\Desktop\c++\Inheritance and Pointers\11.pointer to print array elements.exe

- Output Size: 1.83264636993408 KiB

12. Write a C++ program to create a pointer to an array of character and display its value.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     char arr[] = "Hello";
6     char *ptr = arr;
7
8     cout << "Value of character array pointed by pointer: " << ptr << endl;
9
10    return 0;
11 }
12
```

```
Value of character array pointed by pointer: Hello
-----
Process exited after 2.1 seconds with return value 0
Press any key to continue . . . |
```

Compilation results...

13. Write a C++ program to create a pointer to an array of floats and display its value.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     float arr[] = {1.1, 2.2, 3.3, 4.4, 5.5};
6     float *ptr = arr;
7
8     cout << "Values of float array elements pointed by pointer: ";
9     for (int i = 0; i < 5; ++i) {
10         cout << *(ptr + i) << " ";
11     }
12     cout << endl;
13     return 0;
14 }
```

```
inter: 1.1 2.2 3.3 4.4 5.5
-----
Process exited after 0.3773 seconds with return value 0
Press any key to continue . . .
```

14. Write a C++ program to create a pointer to an object and display its attributes.

```
1 #include <iostream>
2 using namespace std;
3
4 class MyClass {
5 public:
6     int data;
7     MyClass(int d) : data(d) {}
8 };
9
10 int main() {
11     MyClass obj(42);
12     MyClass *ptr = &obj;
13
14     cout << "Value of attribute in object pointed by pointer: " << ptr->data << endl;
15
16     return 0;
17 }
```

```
ter: 42
-----
Process exited after 0.4847 seconds with return value 0
Press any key to continue . . .
```

15. Write a C++ program to create a pointer to a function and call the function using the pointer.

