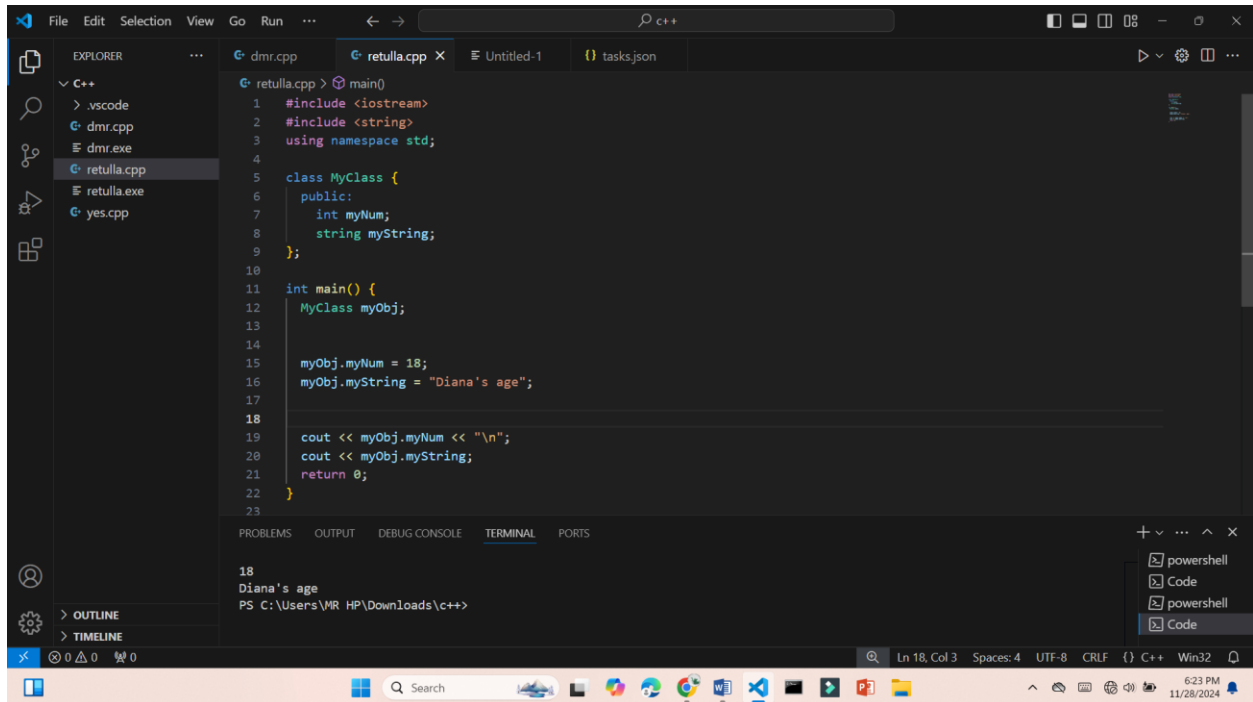


Diana Mae Retulla
BSCPE 1A1

OBJECT-ORIENTED PROGRAMMING

Classes/Objects



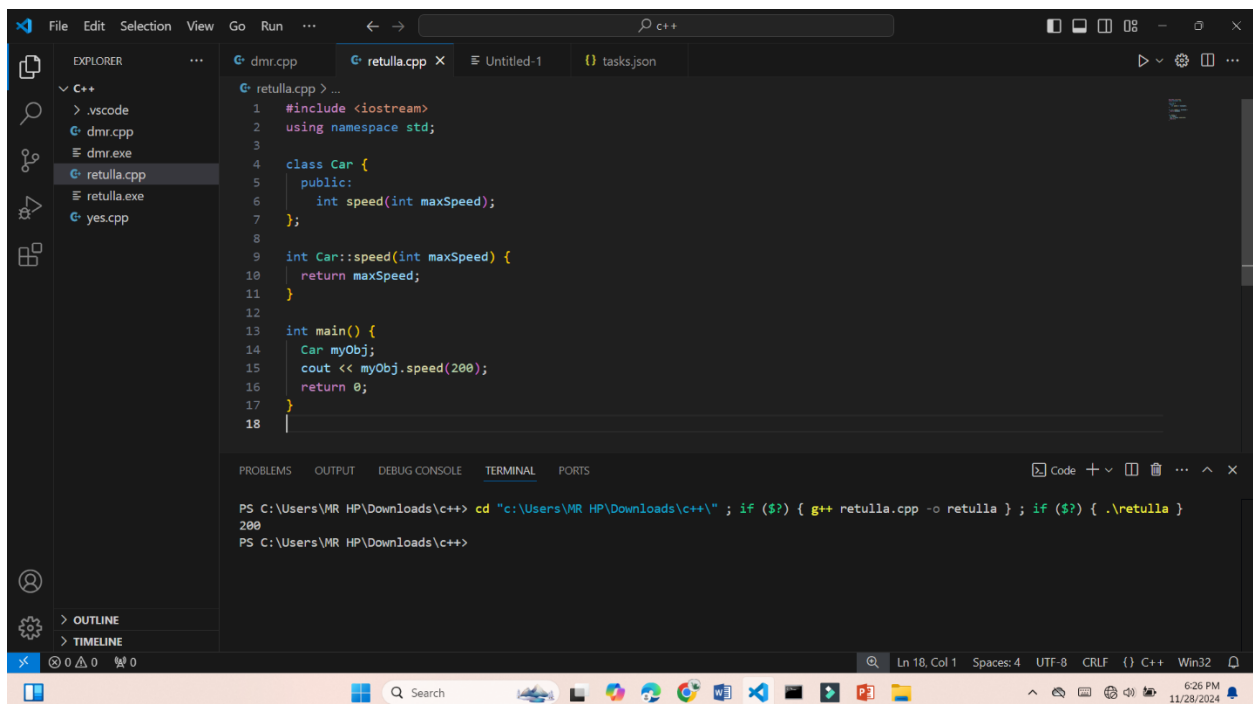
The screenshot shows the Visual Studio Code editor with a C++ file named `retulla.cpp`. The code defines a class `MyClass` with a public attribute `myString` and a method `main()`. The `main()` method creates an object `myObj`, initializes `myNum` to 18 and `myString` to "Diana's age", and prints the values.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 class MyClass {
6 public:
7     int myNum;
8     string myString;
9 };
10
11 int main() {
12     MyClass myObj;
13
14     myObj.myNum = 18;
15     myObj.myString = "Diana's age";
16
17     cout << myObj.myNum << "\n";
18     cout << myObj.myString;
19     return 0;
20 }
```

The terminal output shows the program's execution:

```
18
Diana's age
PS C:\Users\MR HP\Downloads\c++>
```

Class Methods



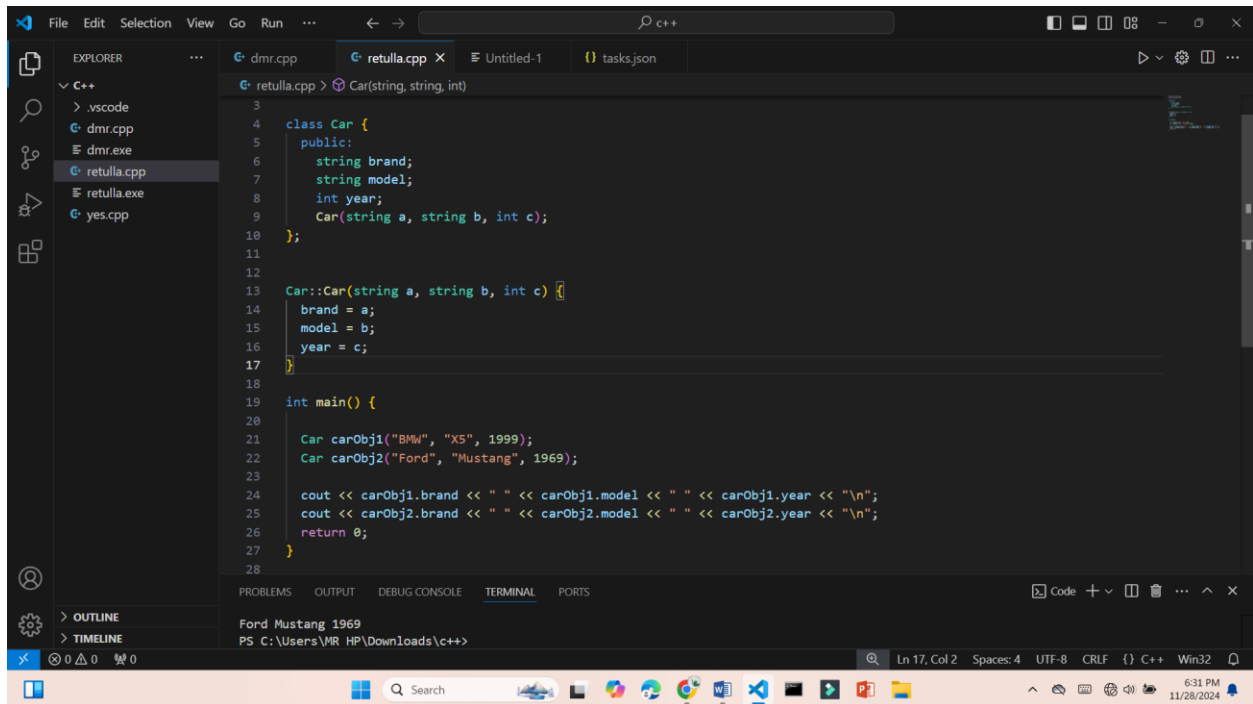
The screenshot shows the Visual Studio Code editor with a C++ file named `retulla.cpp`. The code defines a class `Car` with a public method `speed()`. The `main()` method creates an object `myObj` and calls the `speed()` method with the value 200.

```
1 #include <iostream>
2 using namespace std;
3
4 class Car {
5 public:
6     int speed(int maxSpeed);
7 };
8
9 int Car::speed(int maxSpeed) {
10     return maxSpeed;
11 }
12
13 int main() {
14     Car myObj;
15     cout << myObj.speed(200);
16     return 0;
17 }
```

The terminal output shows the program's execution:

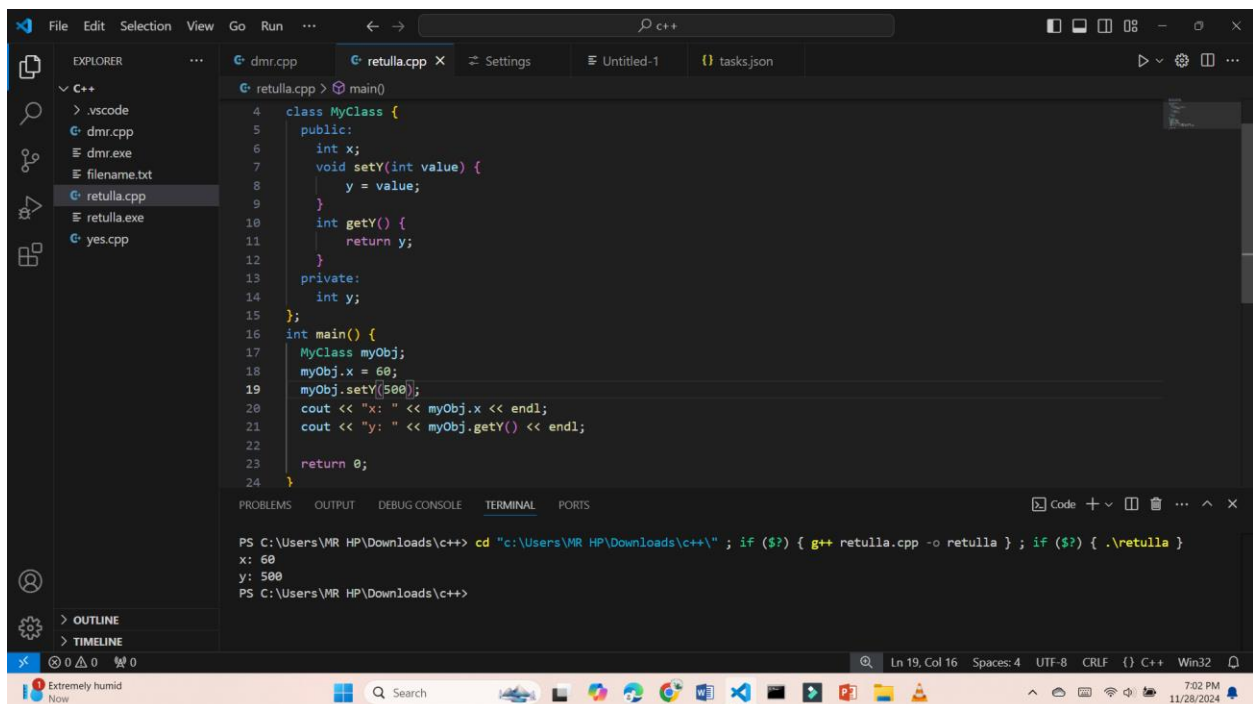
```
PS C:\Users\MR HP\Downloads\c++> cd "c:\Users\MR HP\Downloads\c++\" ; if ($?) { g++ retulla.cpp -o retulla } ; if ($?) { .\retulla }
200
PS C:\Users\MR HP\Downloads\c++>
```

Constructors



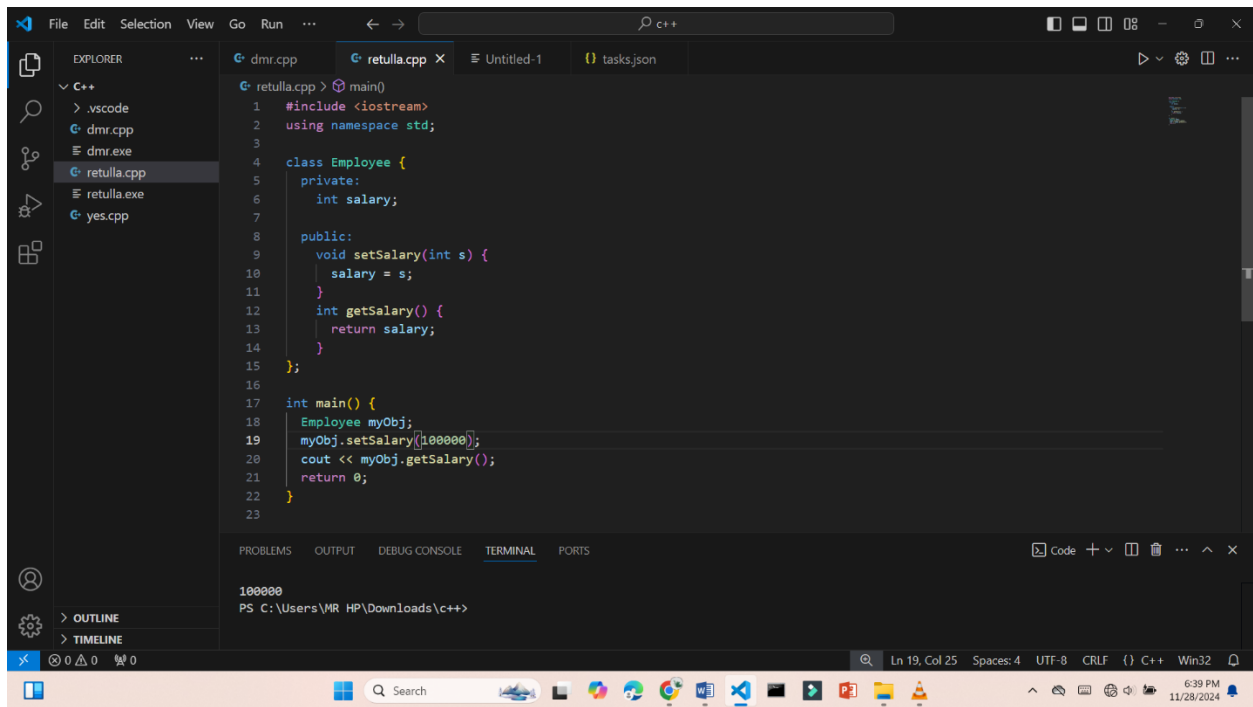
```
File Edit Selection View Go Run ... c++
EXPLORER
  C++
    .vscode
    dmr.cpp
    dmr.exe
    retulla.cpp
    retulla.exe
    yes.cpp
retulla.cpp > Car(string, string, int)
3
4 class Car {
5     public:
6         string brand;
7         string model;
8         int year;
9         Car(string a, string b, int c);
10 };
11
12
13 Car::Car(string a, string b, int c) {
14     brand = a;
15     model = b;
16     year = c;
17 }
18
19 int main() {
20     Car carObj1("BMW", "X5", 1999);
21     Car carObj2("Ford", "Mustang", 1969);
22
23     cout << carObj1.brand << " " << carObj1.model << " " << carObj1.year << "\n";
24     cout << carObj2.brand << " " << carObj2.model << " " << carObj2.year << "\n";
25     return 0;
26 }
27
28
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ford Mustang 1969
PS C:\Users\VMR HP\Downloads\c++>
```

Access Specifiers



```
File Edit Selection View Go Run ... c++
EXPLORER
  C++
    .vscode
    dmr.cpp
    dmr.exe
    filename.txt
    retulla.cpp
    retulla.exe
    yes.cpp
retulla.cpp > main()
4 class MyClass {
5     public:
6         int x;
7         void setY(int value) {
8             y = value;
9         }
10        int getY() {
11            return y;
12        }
13        private:
14            int y;
15    };
16    int main() {
17        MyClass myObj;
18        myObj.x = 60;
19        myObj.setY(500);
20        cout << "x: " << myObj.x << endl;
21        cout << "y: " << myObj.getY() << endl;
22    }
23    return 0;
24 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\VMR HP\Downloads\c++> cd "C:\Users\VMR HP\Downloads\c++\"; if ($?) { g++ retulla.cpp -o retulla }; if ($?) { .\retulla }
x: 60
y: 500
PS C:\Users\VMR HP\Downloads\c++>
```

Encapsulation



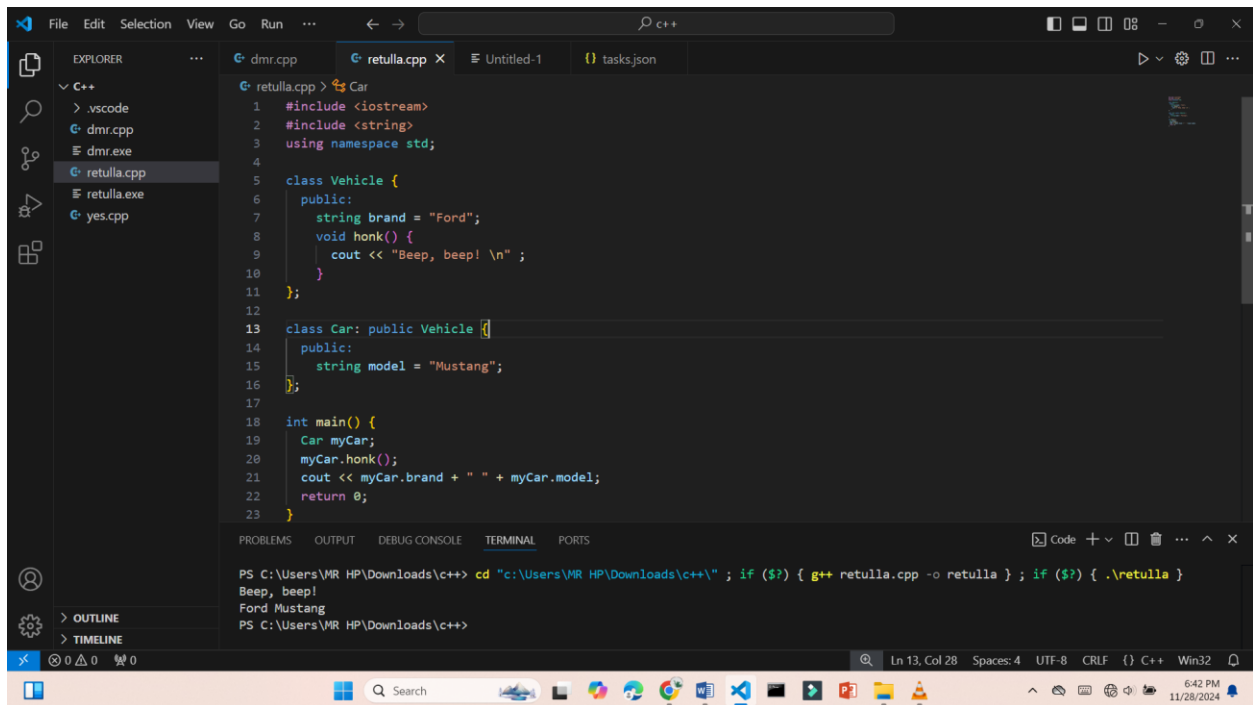
The screenshot shows the Visual Studio Code editor with a C++ file named `retulla.cpp`. The code defines a class `Employee` with a private attribute `salary` and two public methods: `setSalary` and `getSalary`. The `main` function creates an `Employee` object, sets its salary to 100000, and prints it out.

```
1 #include <iostream>
2 using namespace std;
3
4 class Employee {
5     private:
6         int salary;
7
8     public:
9         void setSalary(int s) {
10             salary = s;
11         }
12         int getSalary() {
13             return salary;
14         }
15 };
16
17 int main() {
18     Employee myObj;
19     myObj.setSalary(100000);
20     cout << myObj.getSalary();
21     return 0;
22 }
```

The terminal output shows the result of the program execution:

```
100000
PS C:\Users\MR HP\Downloads\c++>
```

Inheritance



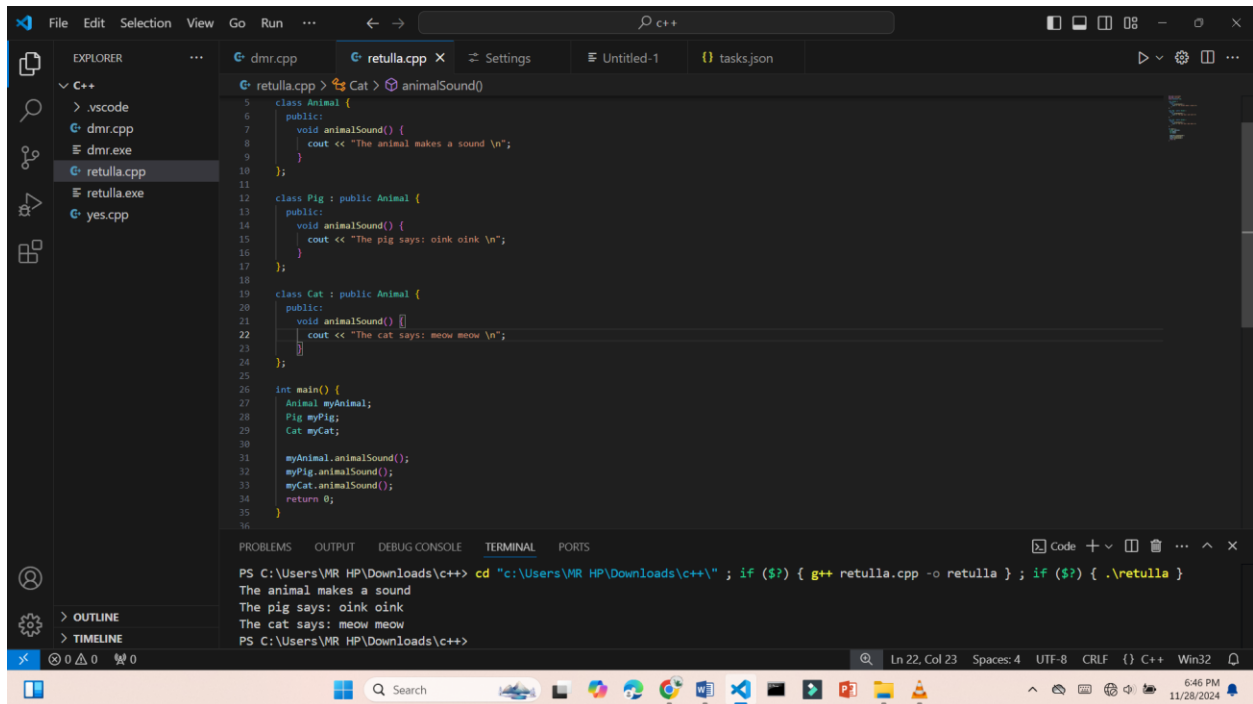
The screenshot shows the Visual Studio Code editor with a C++ file named `retulla.cpp`. The code demonstrates inheritance by creating a base class `Vehicle` and a derived class `Car` that inherits from `Vehicle`. The `Vehicle` class has a public attribute `brand` and a public method `honk`. The `Car` class inherits from `Vehicle` and has a public attribute `model`. The `main` function creates a `Car` object, calls its `honk` method, and prints its brand and model.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 class Vehicle {
6     public:
7         string brand = "Ford";
8         void honk() {
9             cout << "Beep, beep! \n";
10         }
11 };
12
13 class Car: public Vehicle {
14     public:
15         string model = "Mustang";
16 };
17
18 int main() {
19     Car myCar;
20     myCar.honk();
21     cout << myCar.brand + " " + myCar.model;
22     return 0;
23 }
```

The terminal output shows the result of the program execution:

```
PS C:\Users\MR HP\Downloads\c++> cd "c:\Users\MR HP\Downloads\c++\" ; if ($?) { g++ retulla.cpp -o retulla } ; if ($?) { .\retulla }
Beep, beep!
Ford Mustang
PS C:\Users\MR HP\Downloads\c++>
```

Polymorphism



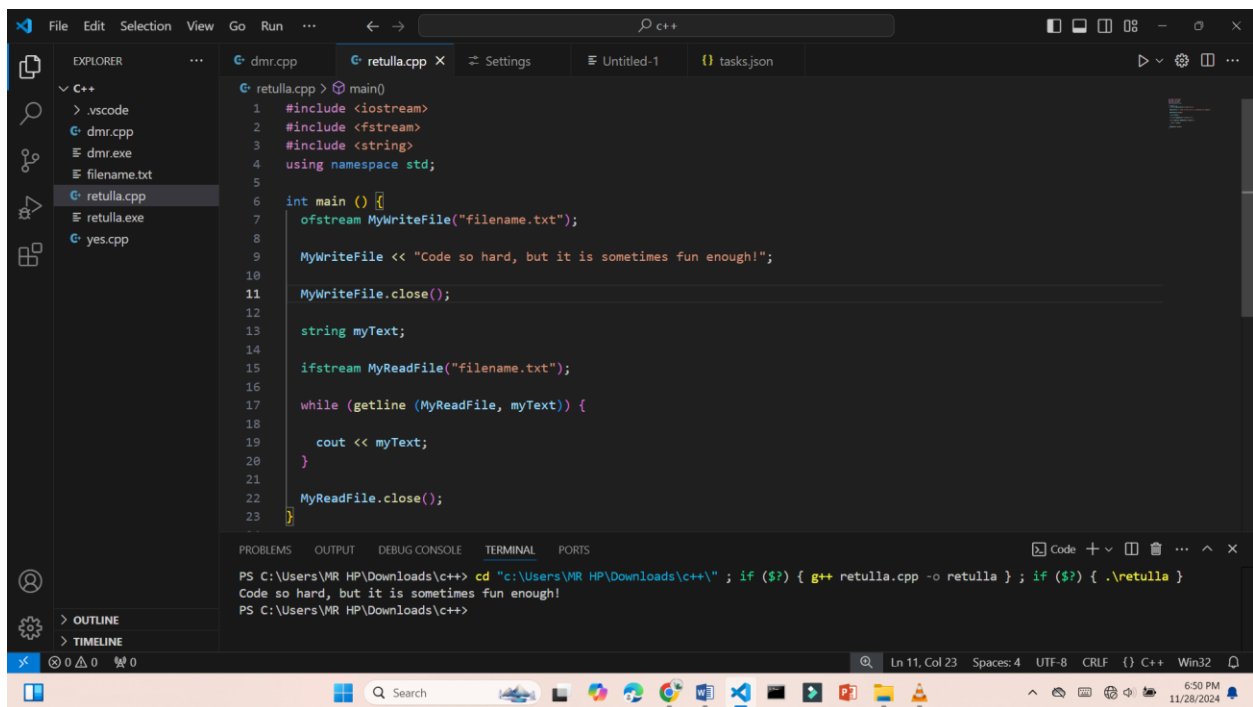
The screenshot shows a Visual Studio Code editor with a C++ project. The Explorer panel on the left shows files: .vscode, dmr.cpp, dmr.exe, retulla.cpp, retulla.exe, and yes.cpp. The main editor displays the code for retulla.cpp, which defines three classes: Animal, Pig, and Cat. Each class has an animalSound() method. The main function creates instances of these classes and calls their animalSound() methods. The output window at the bottom shows the execution results.

```
retulla.cpp > Cat > animalSound()
5  class Animal {
6      public:
7          void animalSound() {
8              cout << "The animal makes a sound \n";
9          }
10 };
11
12 class Pig : public Animal {
13     public:
14         void animalSound() {
15             cout << "The pig says: oink oink \n";
16         }
17 };
18
19 class Cat : public Animal {
20     public:
21         void animalSound() {
22             cout << "The cat says: meow meow \n";
23         }
24 };
25
26 int main() {
27     Animal myAnimal;
28     Pig myPig;
29     Cat myCat;
30
31     myAnimal.animalSound();
32     myPig.animalSound();
33     myCat.animalSound();
34     return 0;
35 }
36
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\MR HP\Downloads\c++> cd "c:\Users\MR HP\Downloads\c++\" ; if ($?) { g++ retulla.cpp -o retulla } ; if ($?) { .\retulla }
The animal makes a sound
The pig says: oink oink
The cat says: meow meow
PS C:\Users\MR HP\Downloads\c++>
```

Files



The screenshot shows a Visual Studio Code editor with a C++ project. The Explorer panel on the left shows files: .vscode, dmr.cpp, dmr.exe, filename.txt, retulla.cpp, retulla.exe, and yes.cpp. The main editor displays the code for retulla.cpp, which includes headers for iostream, fstream, and string. The main function uses ofstream to write to a file and ifstream to read from it. The output window at the bottom shows the execution results.

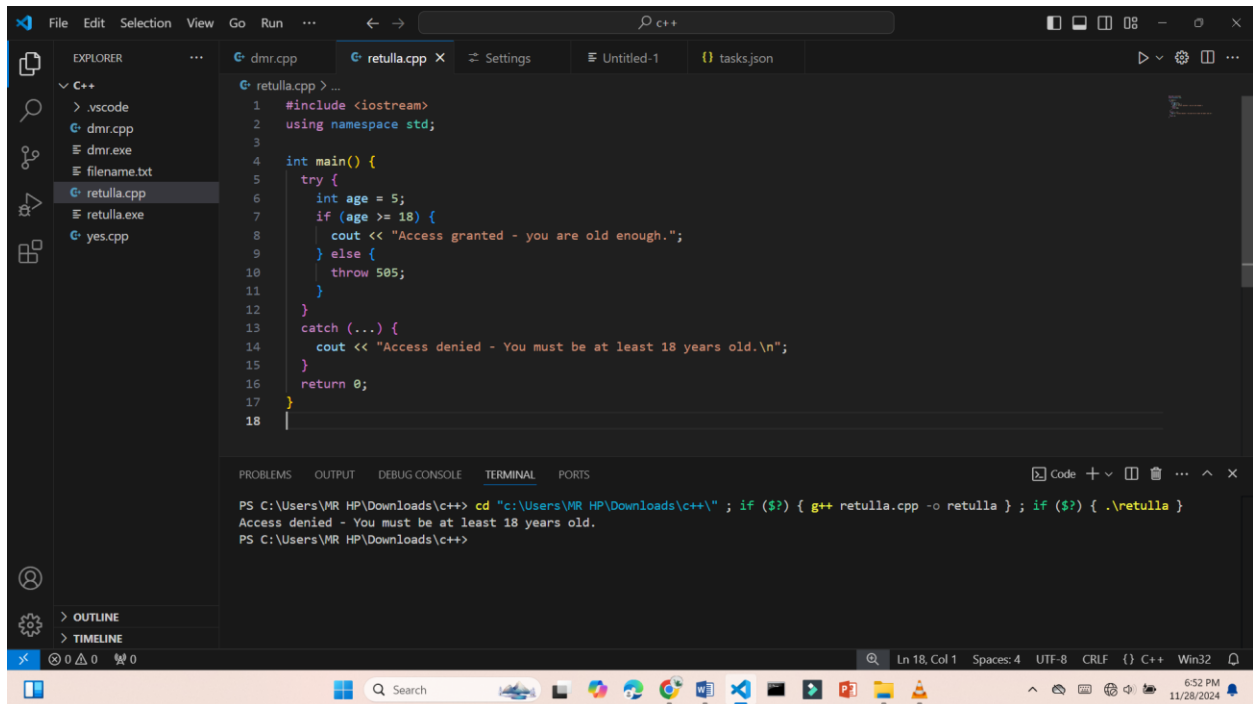
```
retulla.cpp > main()
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  using namespace std;
5
6  int main () {
7      ofstream MyWriteFile("filename.txt");
8
9      MyWriteFile << "Code so hard, but it is sometimes fun enough!";
10
11     MyWriteFile.close();
12
13     string myText;
14
15     ifstream MyReadFile("filename.txt");
16
17     while (getline (MyReadFile, myText)) {
18
19         cout << myText;
20     }
21
22     MyReadFile.close();
23 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\MR HP\Downloads\c++> cd "c:\Users\MR HP\Downloads\c++\" ; if ($?) { g++ retulla.cpp -o retulla } ; if ($?) { .\retulla }
Code so hard, but it is sometimes fun enough!
PS C:\Users\MR HP\Downloads\c++>
```

Exceptions

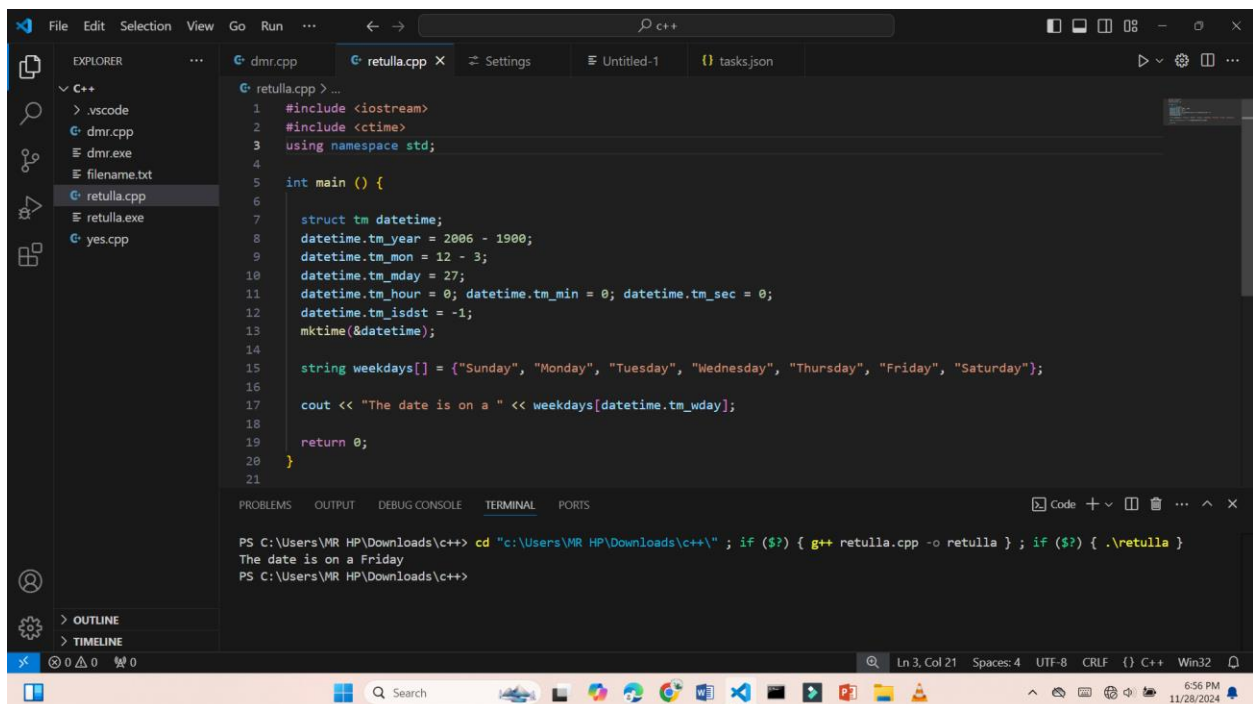


```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     try {
6         int age = 5;
7         if (age >= 18) {
8             cout << "Access granted - you are old enough.";
9         } else {
10            throw 505;
11        }
12    }
13    catch (...) {
14        cout << "Access denied - You must be at least 18 years old.\n";
15    }
16    return 0;
17 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\MR HP\Downloads\c++> cd "c:\Users\MR HP\Downloads\c++\" ; if ($?) { g++ retulla.cpp -o retulla } ; if ($?) { .\retulla }
Access denied - You must be at least 18 years old.
PS C:\Users\MR HP\Downloads\c++>
```

Date and Time



```
1 #include <iostream>
2 #include <ctime>
3 using namespace std;
4
5 int main () {
6
7     struct tm datetime;
8     datetime.tm_year = 2006 - 1900;
9     datetime.tm_mon = 12 - 3;
10    datetime.tm_mday = 27;
11    datetime.tm_hour = 0; datetime.tm_min = 0; datetime.tm_sec = 0;
12    datetime.tm_isdst = -1;
13    mktime(&datetime);
14
15    string weekdays[] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
16
17    cout << "The date is on a " << weekdays[datetime.tm_wday];
18
19    return 0;
20 }
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\MR HP\Downloads\c++> cd "c:\Users\MR HP\Downloads\c++\" ; if ($?) { g++ retulla.cpp -o retulla } ; if ($?) { .\retulla }
The date is on a Friday
PS C:\Users\MR HP\Downloads\c++>
```