TEST 4

**NAME: MADHAN S**

**Regno:192221052**

**course code:DSA0163**

**course name:Object oriented c++**

**1. create a base function called person with virtual function greet() and it has two derived function student and teacher and use the greet()function**

#include <iostream>

using namespace std;

class person

{

public:

virtual void greet()

{

cout<<"Greet the student and teacher \n ";

}

};

class student : public person

{

public:

void greet()

{

cout<<"Good morning teacher \n";

}

};

class teacher : public person

{

public:

void greet()

{

cout<<"Good morning Student \n";

}

};

int main()

{

person P;

student s;

teacher t;

person \*p1=&P;

person \*p2=&s;

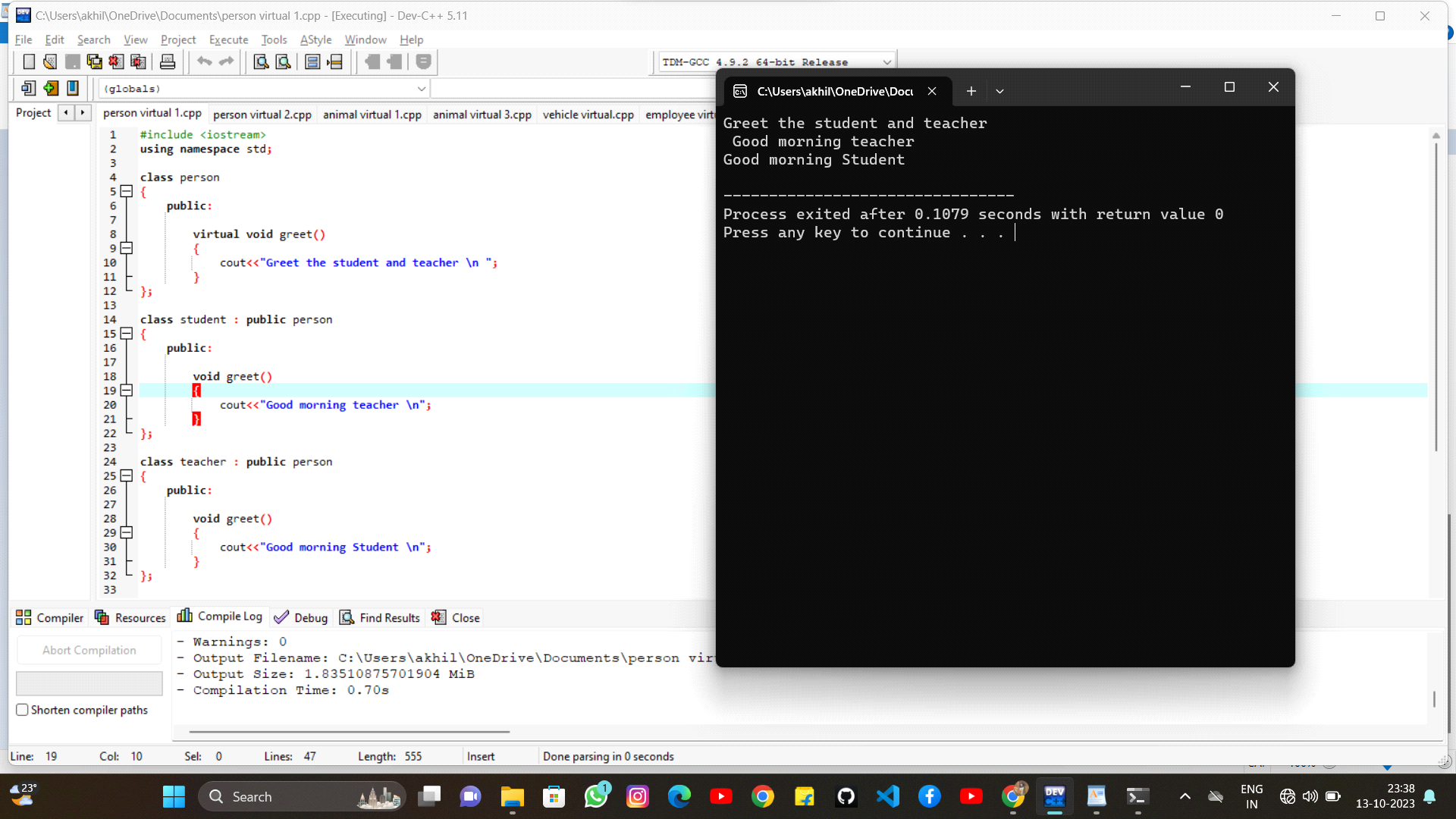
person \*p3=&t;

p1->greet();

p2->greet();

p3->greet();

}



**2. create a base function called person with virtual function work() and it has two derived function manager and employee and use the work()function**

#include <iostream >

using namespace std;

class person

{

public:

virtual void work()

{

}

};

class employee : public person

{

public:

void work()

{

cout<<"Employee is working \n";

}

};

class manager : public person

{

public:

void work()

{

cout<<"Manager is working \n\";

}

};

int main()

{

person p;

employee e;

manager m;

person \*p1=&p;

person \*p2=&e;

person \*p3=&m;

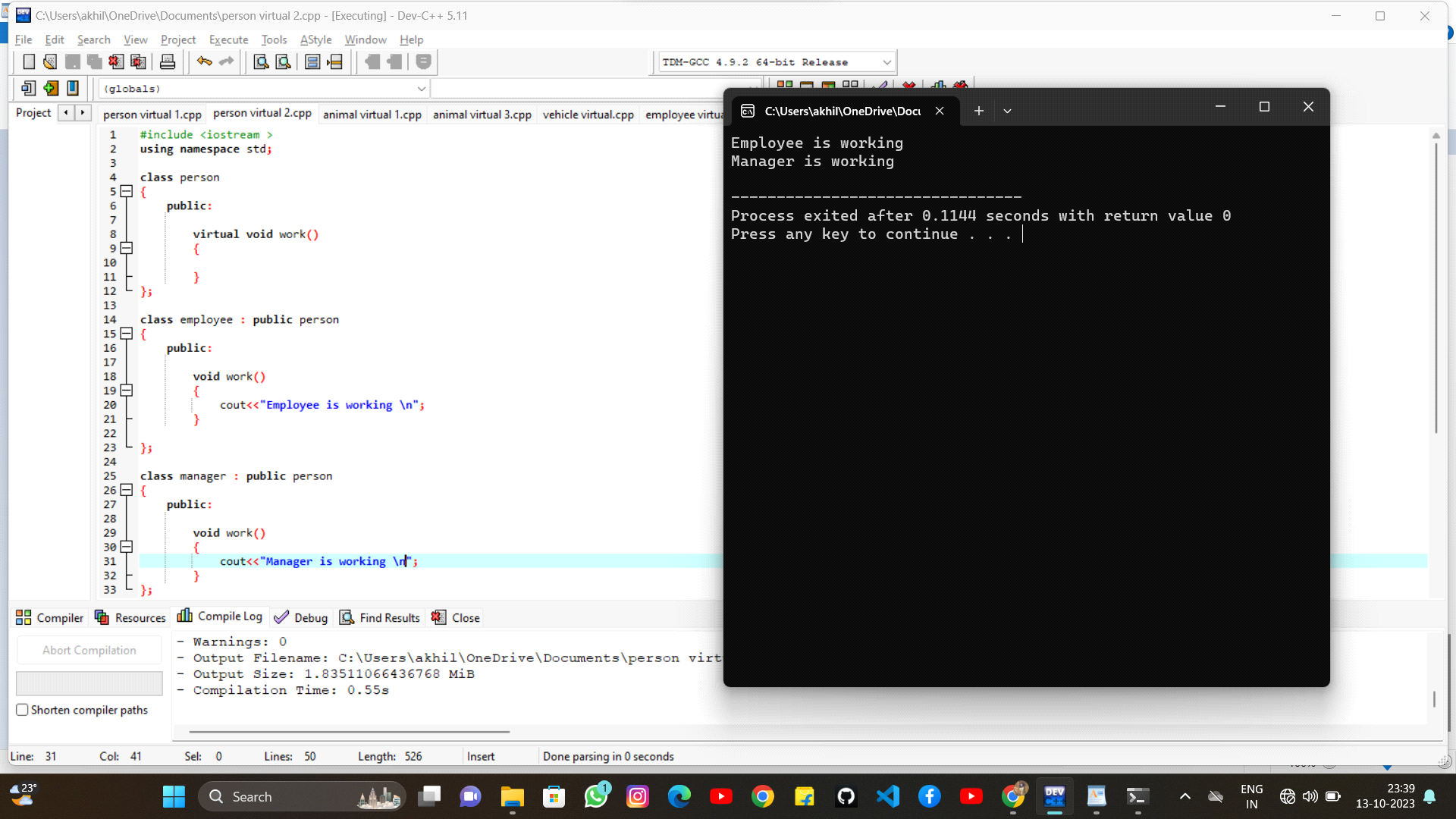
p1->work();

p2->work();

p3->work();

return 0;

}



**3. create a base function called animal with virtual function eat() and it has two derived function herbivore and carnivore and use the greet()function**

#include <iostream>

using namespace std;

class animal

{

public:

virtual void eat()

{

cout<<"Animals are two types \n";

}

};

class herbivore : public animal

{

public:

void eat()

{

cout<<"herbivore are plants eating animal \n";

}

};

class carnivore :public animal

{

public:

void eat()

{

cout<<"Carnivore are animals eating animal \n";

}

};

int main()

{

animal a;

herbivore h;

carnivore c;

animal \*a1=&a;

animal \*a2=&h;

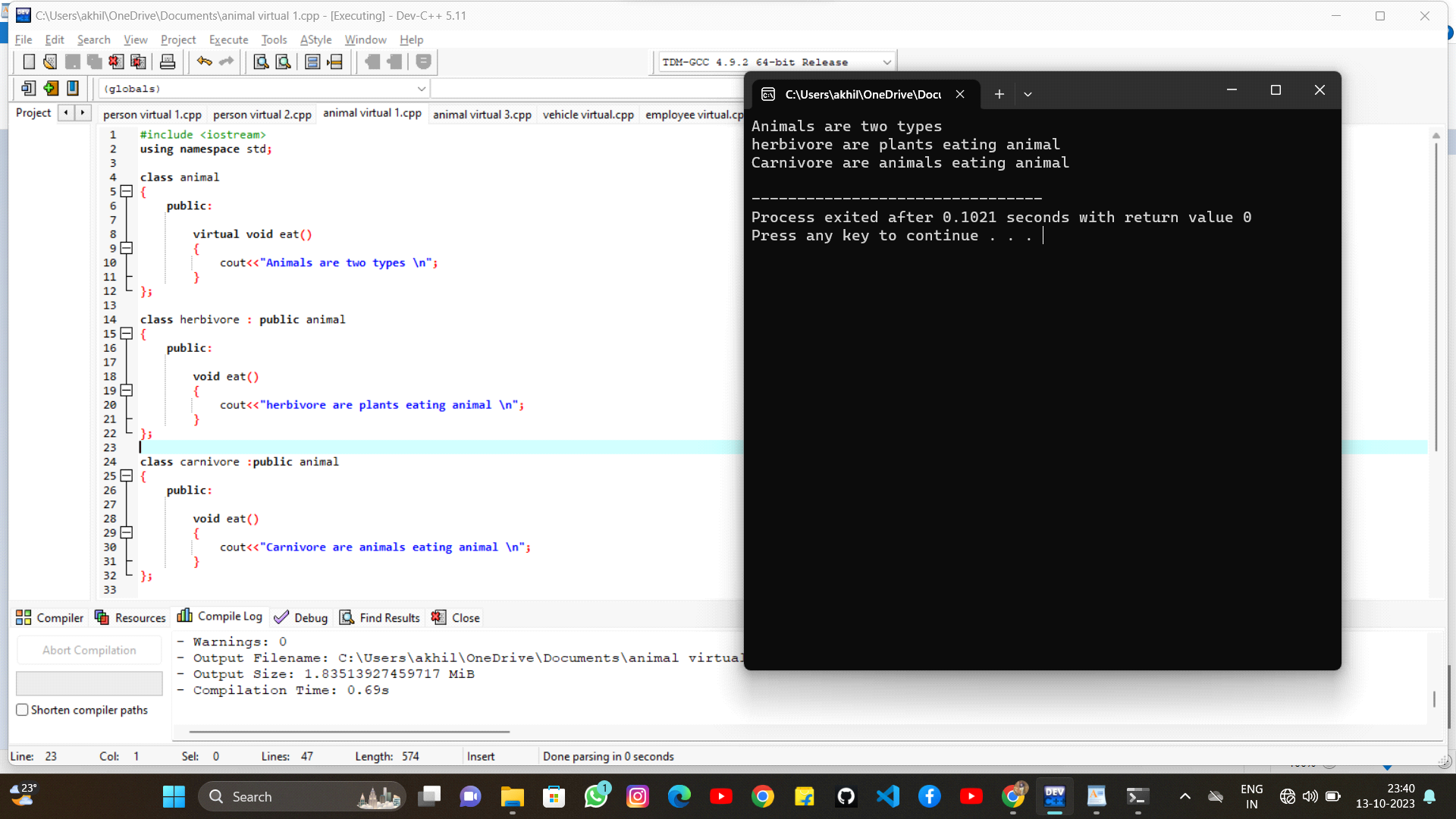
animal \*a3=&c;

a1->eat();

a2->eat();

a3->eat();

}



**4. create a base function called animal with virtual function speak() and it has two derived function caat and dog and use the speak()function**

#include <iostream>

using namespace std;

class animal

{

public:

virtual void speak()

{

cout<<"Animals are two types \n";

}

};

class cat : public animal

{

public:

void speak()

{

cout<<"cat will speak meow meow....\n";

}

};

class dog :public animal

{

public:

void speak()

{

cout<<"dog will speak boww boww..... \n";

}

};

int main()

{

animal a;

cat c;

dog d;

animal \*a1=&a;

animal \*a2=&c;

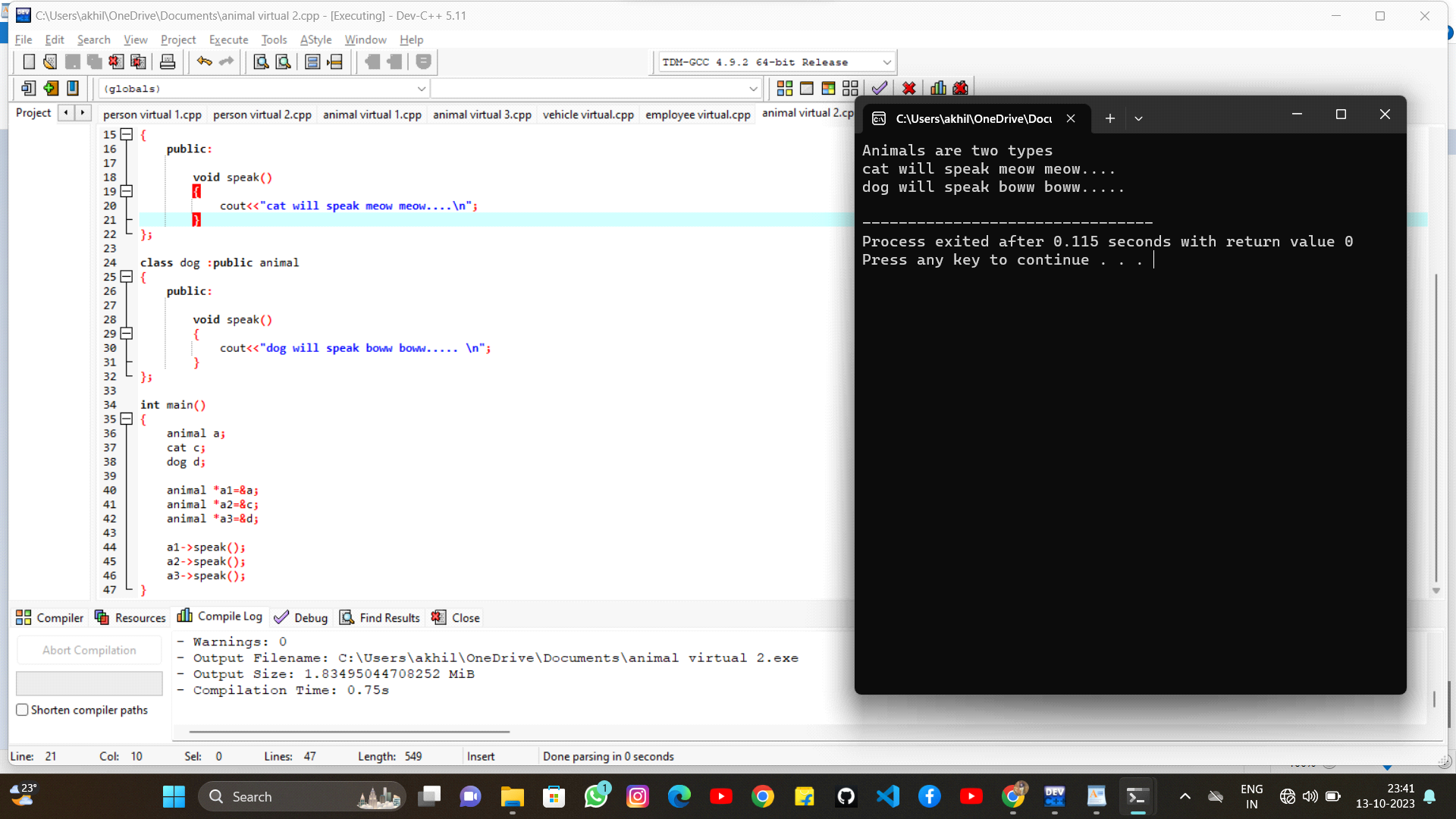
animal \*a3=&d;

a1->speak();

a2->speak();

a3->speak();

}



**5. create a base function called animal with virtual function move() and it has two derived function bird and fish and use the move()function**

#include <iostream>

using namespace std;

class animal

{

public:

virtual void move()

{

cout<<"Animals are two types \n";

}

};

class bird : public animal

{

public:

void move()

{

cout<<"birds will fly in sky \n";

}

};

class fish :public animal

{

public:

void move()

{

cout<<"fish will swim in water \n";

}

};

int main()

{

animal a;

bird b;

fish f;

animal \*a1=&a;

animal \*a2=&b;

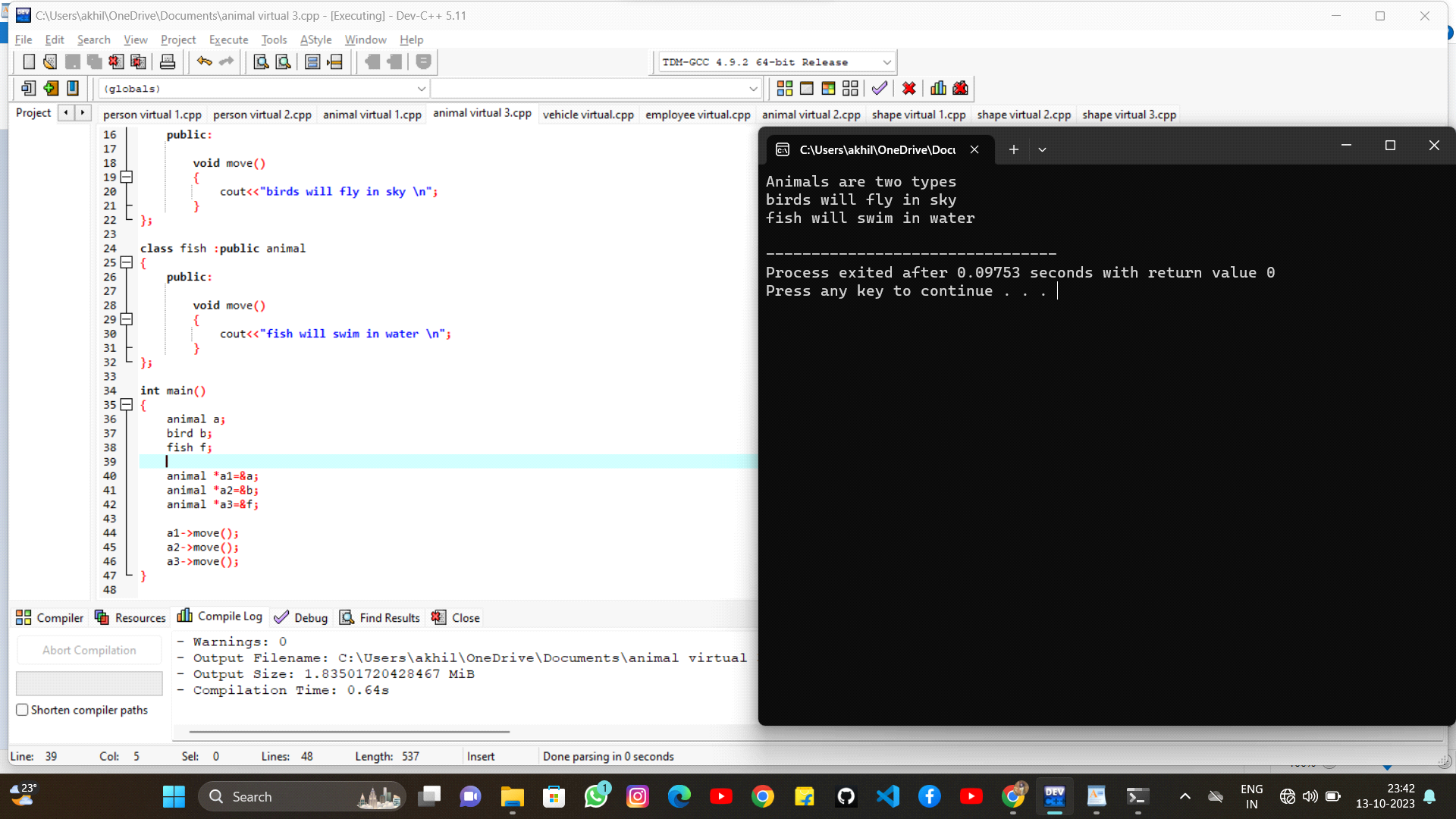
animal \*a3=&f;

a1->move();

a2->move();

a3->move();

}



**6. create a base function called vehicle with virtual function drive() and it has two derived function car and truck and use the drive()function**

#include <iostream>

using namespace std;

class vehicle

{

public:

virtual void drive()

{

cout<<"Vehicles are two types \n";

}

};

class car : public vehicle

{

public:

void drive()

{

cout<<"Car has less fuel \n";

}

};

class truck : public vehicle

{

public:

void drive()

{

cout<<"Truck has a more fuel \n";

}

};

int main()

{

vehicle v;

car c;

truck t;

vehicle \*v1=&v;

vehicle \*v2=&c;

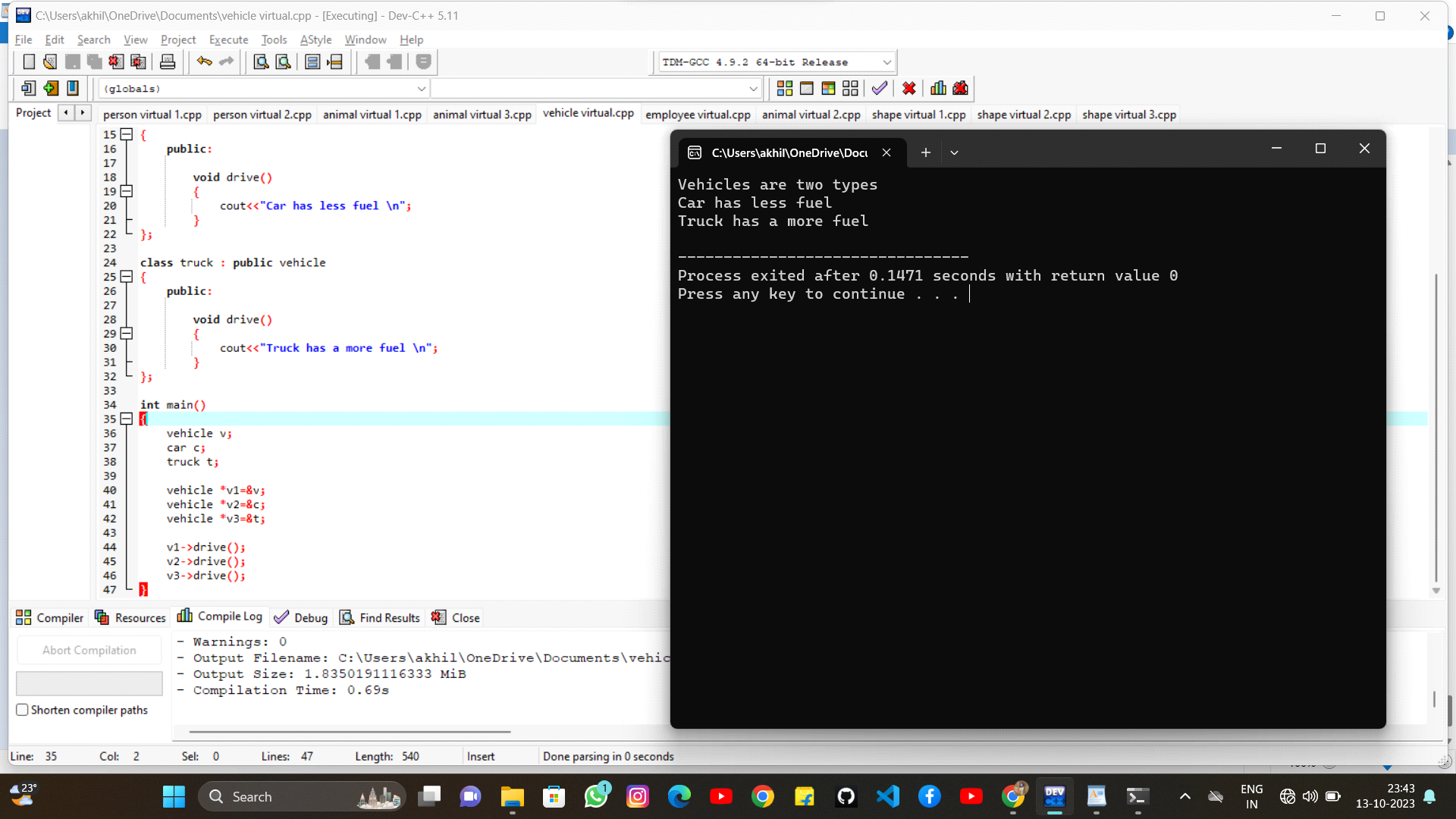
vehicle \*v3=&t;

v1->drive();

v2->drive();

v3->drive();

}



**7. create a base function called Employee with virtual function calculatepay() and it has two derived function manager and engineeer and use the calculatepay()function**

#include <iostream>

using namespace std;

class employee

{

public:

virtual void calculatepay()

{

cout<<"calculacte pay for both \n";

}

};

class manager : public employee

{

public:

void calculatepay()

{

cout<<"Manger has 100k \n";

}

};

class engineer : public manager

{

public:

void calculatepay()

{

cout<<"Engineer has 50k\n";

}

};

int main()

{

employee e;

manager m;

engineer E;

employee \*e1=&e;

employee \*e2=&m;

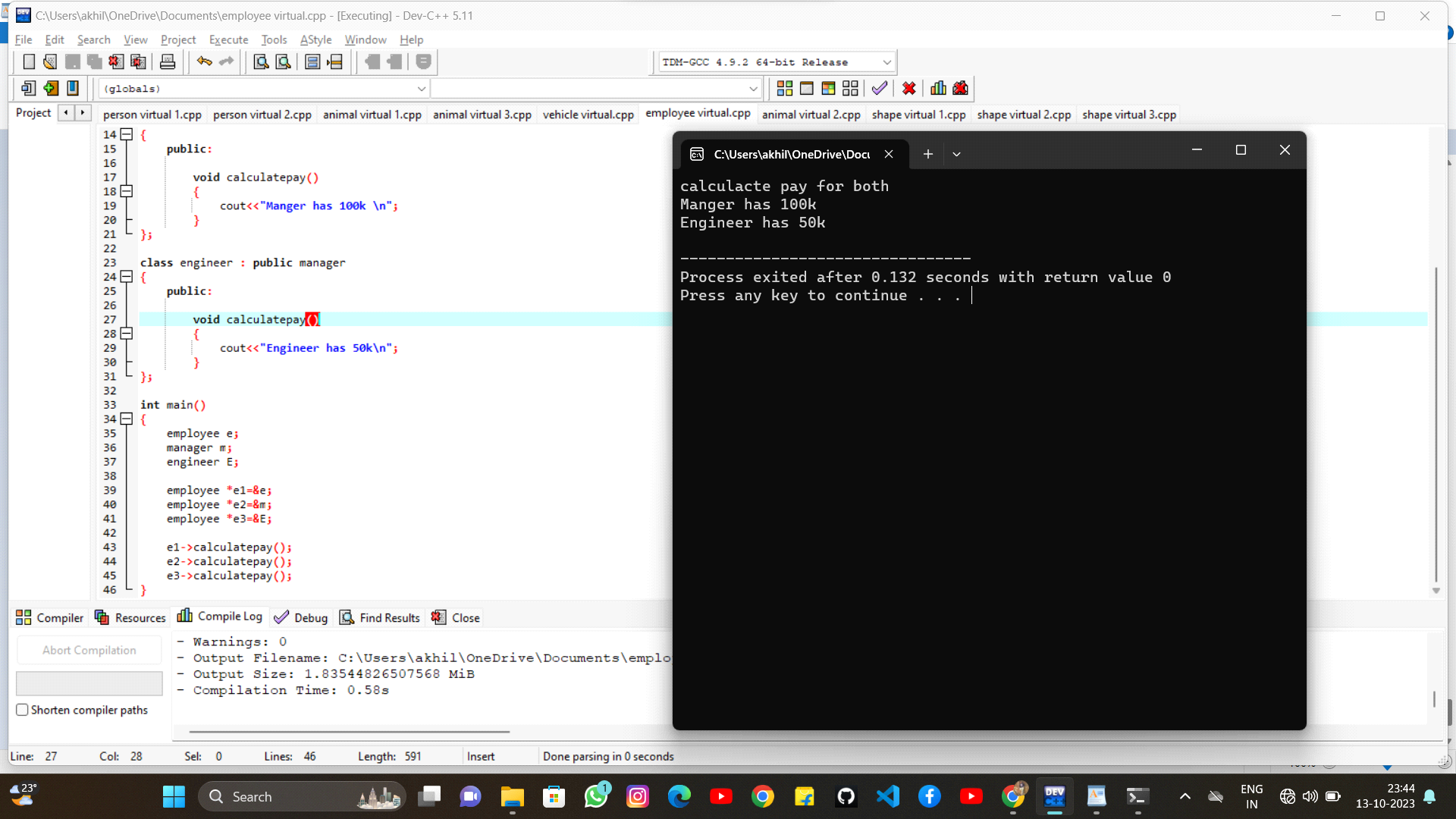
employee \*e3=&E;

e1->calculatepay();

e2->calculatepay();

e3->calculatepay();

}



**8. create a base function called shape with virtual function area() and perimeter and it has two derived rectangle and triangle and use the area() and perimeter function**

#include <iostream>

using namespace std;

class shape

{

public:

virtual void area()

{

}

virtual void perimeter()

{

}

};

class rectangle : public shape

{

public:

int length;

int width;

void read()

{

cout<<"ENter the value of length";

cin>>length;

cout<<"Enter the value of width";

cin>>width;

}

void area()

{

cout<<"The area of rectangle ="<<length\*width<<"\n";

}

void perimeter()

{

cout<<"The perimeter of rectangle ="<<2\*(length+width)<<"\n";

}

};

class triangle : public shape

{

public:

int height;

int base;

void read1()

{

cout<<"Enter the value of height";

cin>>height;

cout<<"Enter the value of base";

cin>>base;

}

void area()

{

cout<<"area of triangle ="<<0.5\*height\*base<<"\n";

}

void perimeter()

{

cout<<"Perimeter of triangle ="<<base+base+base<<"\n";

}

};

int main()

{

shape s;

rectangle r;

triangle t;

r.read();

t.read1();

shape \*s1=&s;

shape \*s2=&r;

shape \*s3=&t;

s1->area();

s1->perimeter();

s2->area();

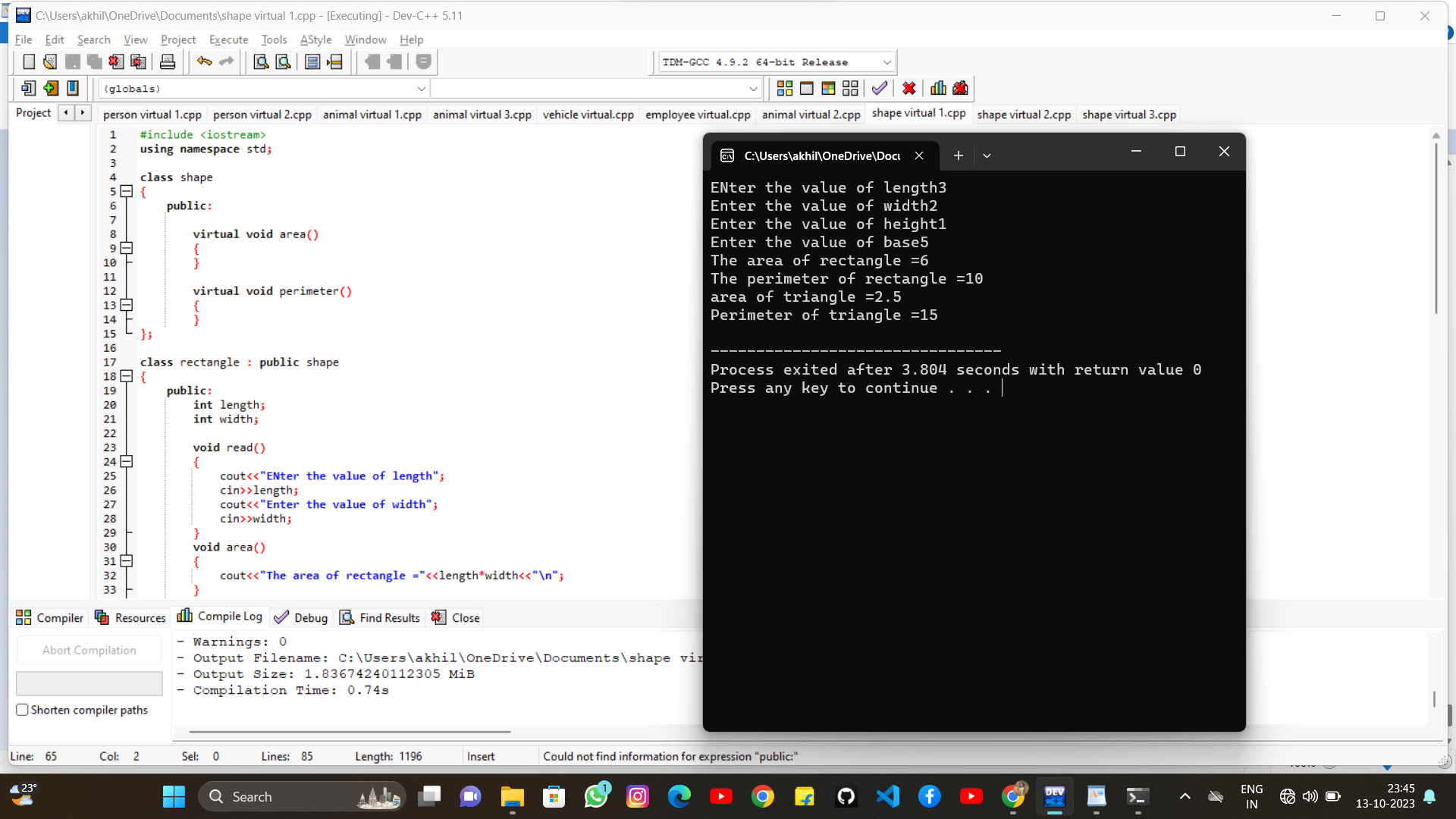
s2->perimeter();

s3->area();

s3->perimeter();

return 0;

}



**9. create a base function called shape with virtual function area() and volume() and it has two derived sphere and cylinder and use the area() and volume() function**

#include <iostream>

using namespace std;

class shape

{

public :

virtual void area()

{

//cout<<"Animals are two types \n";

}

virtual void volume()

{

}

};

class sphere: public shape

{

public :

int radius;

int pi;

void read()

{

radius=5;

pi=3.17;

}

void area()

{

cout<<"Area of the sphere = "<<4\*pi\*radius\*radius<<"\n";

}

void volume()

{

cout<<"perimeter of sphere = "<<2\*pi\*radius<<"\n";

}

};

class cylinder : public shape

{

public:

int radius;

int height;

int pi;

void read1()

{

radius=5;

height=10;

pi=3.14;

}

void area()

{

cout<< "area of cylinder = "<<2\*radius\*height<<"\n";

}

void volume()

{

cout<<"volume of the cylinder ="<<(2\*radius\*height)<<"\n";

}

};

int main()

{

shape s;

sphere S;

cylinder C;

S.read();

C.read1();

shape \*a= &s;

shape \*b= &S;

shape \*c= &C;

a->area();

a->volume();

b->area();

b->volume();

c->area();

c->volume();

return 0;

}



**10. create a base function called shape with virtual function area() and it has two derived rectangle and circle and use the area() function**

#include <iostream>

using namespace std;

class shape

{

public :

virtual void area()

{

//cout<<"Animals are two types \n";

}

};

class rectangle : public shape

{

public :

int length;

int width;

void read()

{

length=10;

width=5;

}

void area()

{

cout<<"Area of the rectangle = "<<length\*width<<"\n";

}

};

class circle : public shape

{

public:

int radius;

int pi;

void read1()

{

radius=10;

pi=3.14;

}

void area()

{

cout<< "area of circle = "<<2\*pi\*radius\*radius<<"\n";

}

};

int main()

{

shape s;

rectangle r;

circle C;

r.read();

C.read1();

shape \*a= &s;

shape \*b= &r;

shape \*c= &C;

a->area();

b->area();

c->area();

return 0;

}

