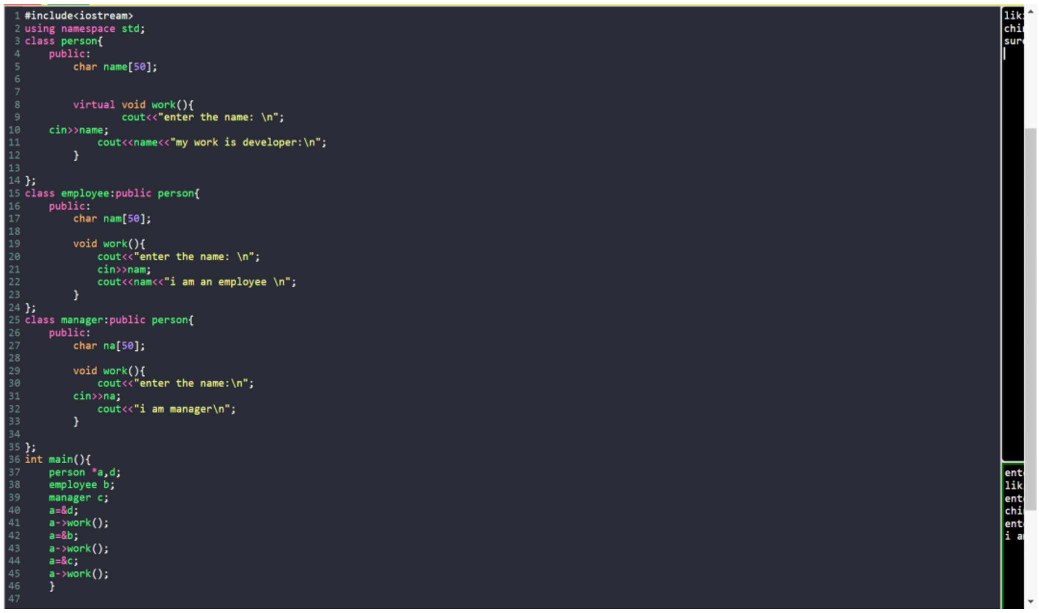
POLYMORPHISM

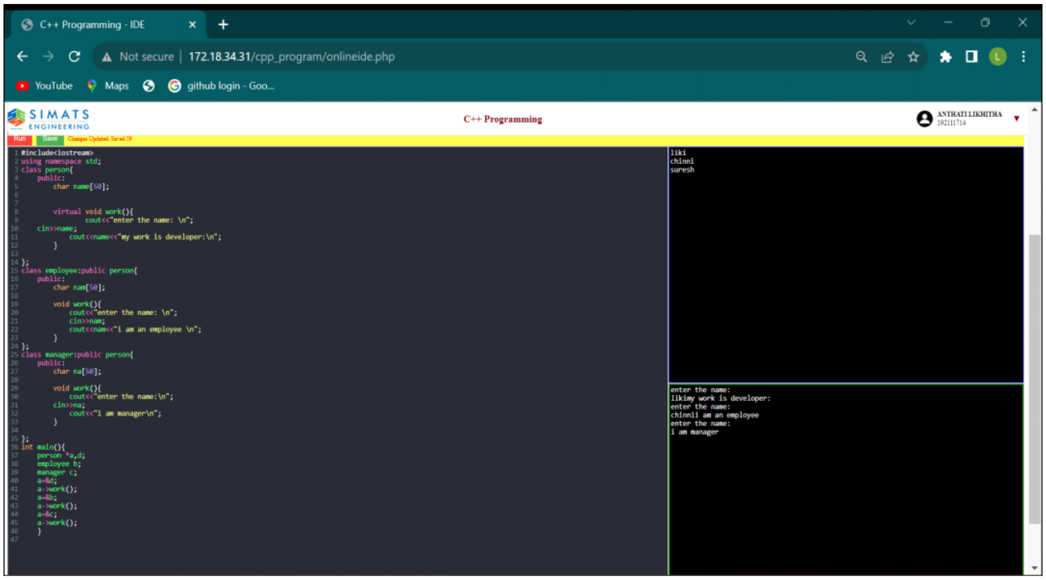
1.Create a base class called Person with a virtual function work(). Derive two classes Employee and Manager from the base class. Implement the work() function for each class.



```
#include<iostream>
using namespace std;
class person{
    public:
        char name[50];


        virtual void work(){
                cout<<"enter the name: \n";
    cin>>name;
            cout<<name<<"my work is developer:\n";
        }

};
class employee:public person{
    public:
        char nam[50];

        void work(){
            cout<<"enter the name: \n";
            cin>>nam;
            cout<<nam<<"i am an employee \n";
        }
};
class manager:public person{
    public:
        char na[50];

        void work(){
            cout<<"enter the name:\n";
        cin>>na;
            cout<<"i am manager\n";
        }

};
int main(){
    person *a,d;
    employee b;
    manager c;
    a=&d;
    a->work();
    a=&b;
    a->work();
    a=&c;
    a->work();
    }
```

```
lik
chi
sur
```

```
ent
lik
ent
chi
ent
i a
```

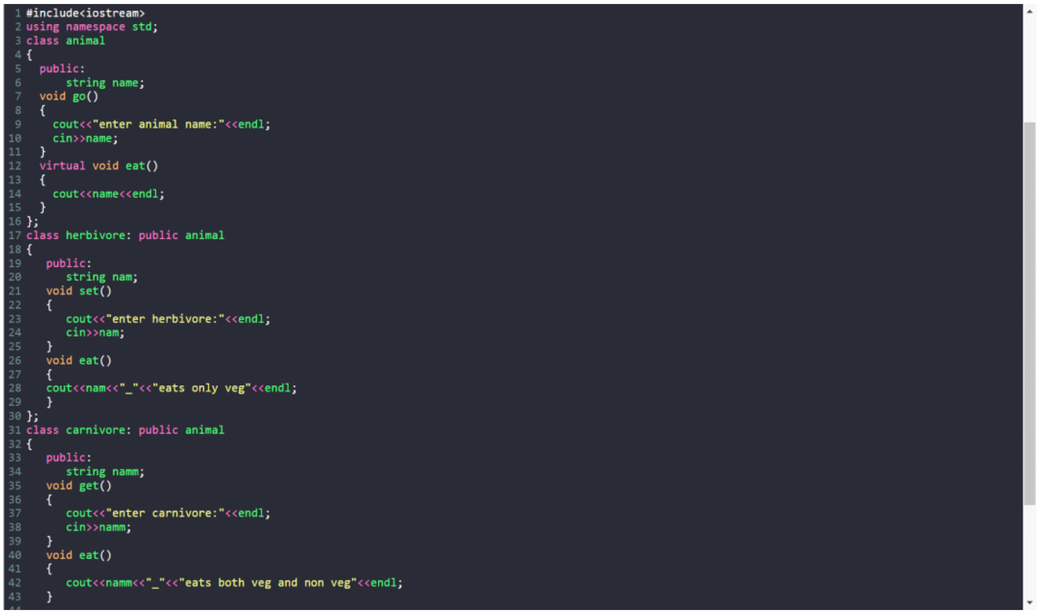


```
#include<iostream>
using namespace std;
class person{
    public:
        char name[50];


        virtual void work(){
                cout<<"enter the name: \n";
    cin>>name;
            cout<<name<<"my work is developer:\n";
        }

};
class employee:public person{
    public:
        char nam[50];

        void work(){
            cout<<"enter the name: \n";
            cin>>nam;
            cout<<nam<<"i am an employee \n";
        }
};
class manager:public person{
    public:
        char na[50];

        void work(){
            cout<<"enter the name:\n";
        cin>>na;
            cout<<"i am manager\n";
        }

};
int main(){
    person *a,d;
    employee b;
    manager c;
    a=&d;
    a->work();
    a=&b;
    a->work();
    a=&c;
    a->work();
    }
```

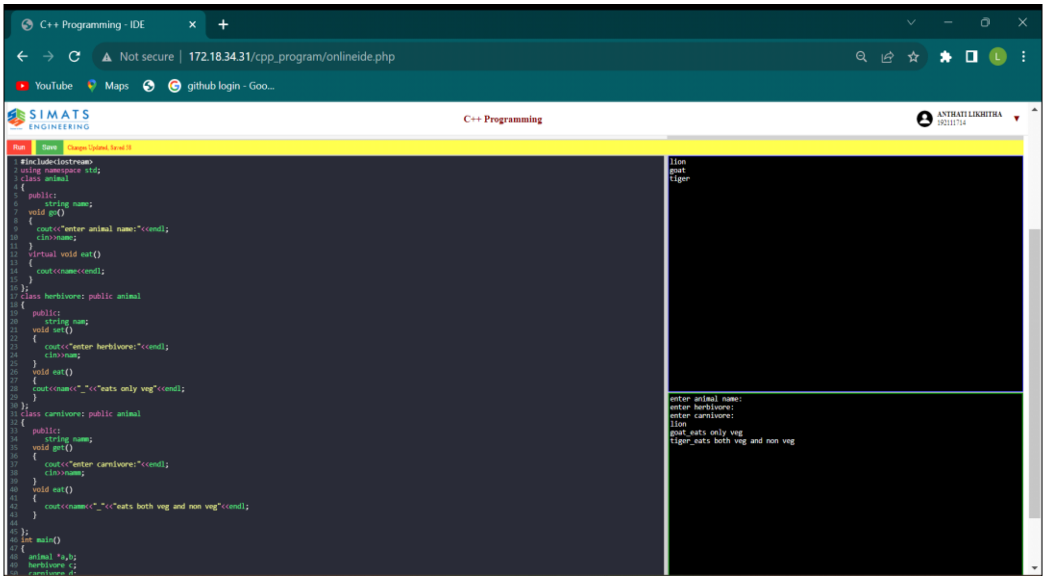
```
liki
chinni
suresh
```

```
enter the name:
likimy work is developer:
enter the name:
chinnii am an employee
enter the name:
i am manager
```

2. Create a base class called Animal with a virtual function eat(). Derive two classes Herbivore and Carnivore from the base class. Implement the eat() function for each class.



```cpp
#include<iostream>
using namespace std;
class animal
{
  public:
      string name;
  void go()
  {
    cout<<"enter animal name:"<<endl;
    cin>>name;
  }
  virtual void eat()
  {
    cout<<name<<endl;
  }
};
class herbivore: public animal
{
   public:
      string nam;
   void set()
   {
      cout<<"enter herbivore:"<<endl;
      cin>>nam;
   }
   void eat()
   {
   cout<<nam<<"_"<<"eats only veg"<<endl;
   }
};
class carnivore: public animal
{
   public:
      string namm;
   void get()
   {
      cout<<"enter carnivore:"<<endl;
      cin>>namm;
   }
   void eat()
   {
      cout<<namm<<"_"<<"eats both veg and non veg"<<endl;
   }
```

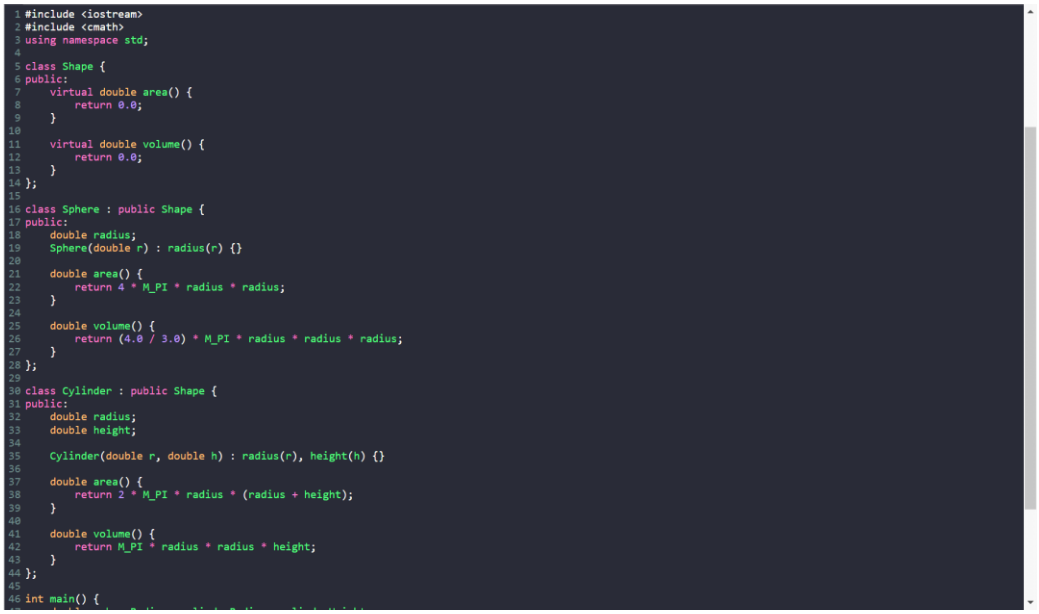

C++ Programming - IDE
Not secure | 172.18.34.31/cpp_program/onlineide.php
YouTube
Maps
github login - Goo...
SIMATS ENGINEERING
C++ Programming
ANTHATI LIKHITHA 192111714
Run
Save
Changes Updated, Saved 58

```cpp
#include<iostream>
using namespace std;
class animal
{
  public:
      string name;
  void go()
  {
    cout<<"enter animal name:"<<endl;
    cin>>name;
  }
  virtual void eat()
  {
    cout<<name<<endl;
  }
};
class herbivore: public animal
{
   public:
      string nam;
   void set()
   {
      cout<<"enter herbivore:"<<endl;
      cin>>nam;
   }
   void eat()
   {
   cout<<nam<<"_"<<"eats only veg"<<endl;
   }
};
class carnivore: public animal
{
   public:
      string namm;
   void get()
   {
      cout<<"enter carnivore:"<<endl;
      cin>>namm;
   }
   void eat()
   {
      cout<<namm<<"_"<<"eats both veg and non veg"<<endl;
   }

};
int main()
{
  animal *a,b;
  herbivore c;
  carnivore d;
```

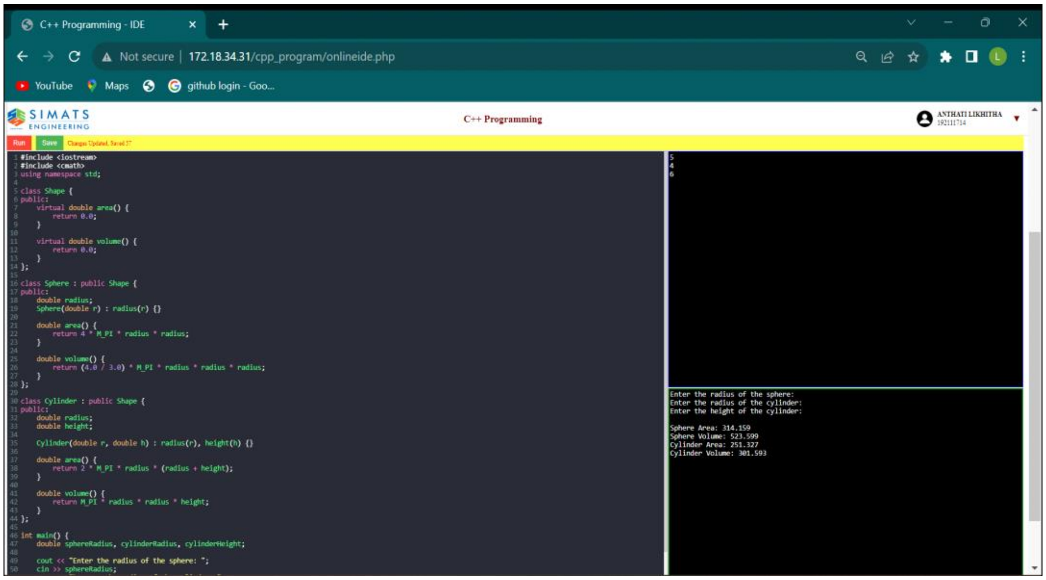
```
lion
goat
tiger
```

```
enter animal name:
enter herbivore:
enter carnivore:
lion
goat_eats only veg
tiger_eats both veg and non veg
```

3.. Create a base class called Shape with virtual functions area() and volume(). Derive two classes Sphere and Cylinder from the base class. Implement the area() and volume() functions for each class.



```cpp
#include <iostream>
#include <cmath>
using namespace std;

class Shape {
public:
    virtual double area() {
        return 0.0;
    }

    virtual double volume() {
        return 0.0;
    }
};

class Sphere : public Shape {
public:
    double radius;
    Sphere(double r) : radius(r) {}

    double area() {
        return 4 * M_PI * radius * radius;
    }

    double volume() {
        return (4.0 / 3.0) * M_PI * radius * radius * radius;
    }
};

class Cylinder : public Shape {
public:
    double radius;
    double height;

    Cylinder(double r, double h) : radius(r), height(h) {}

    double area() {
        return 2 * M_PI * radius * (radius + height);
    }

    double volume() {
        return M_PI * radius * radius * height;
    }
};

int main() {
```



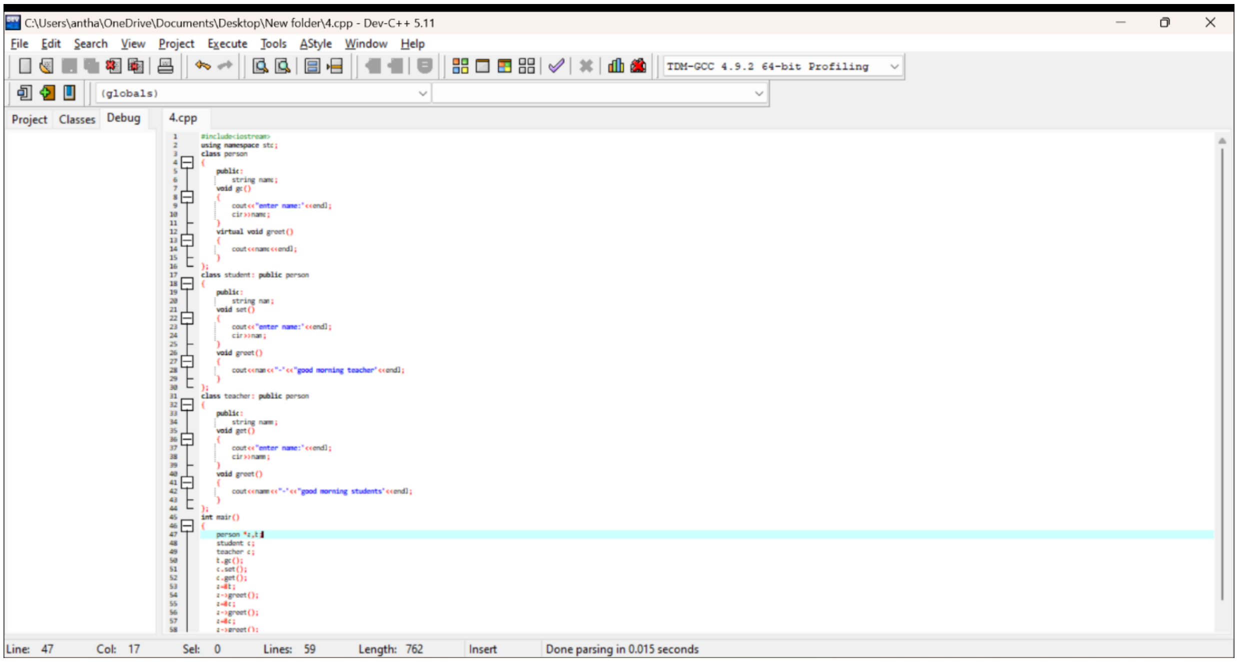
```cpp
#include <iostream>
#include <cmath>
using namespace std;

class Shape {
public:
    virtual double area() {
        return 0.0;
    }

    virtual double volume() {
        return 0.0;
    }
};

class Sphere : public Shape {
public:
    double radius;
    Sphere(double r) : radius(r) {}

    double area() {
        return 4 * M_PI * radius * radius;
    }

    double volume() {
        return (4.0 / 3.0) * M_PI * radius * radius * radius;
    }
};

class Cylinder : public Shape {
public:
    double radius;
    double height;

    Cylinder(double r, double h) : radius(r), height(h) {}

    double area() {
        return 2 * M_PI * radius * (radius + height);
    }

    double volume() {
        return M_PI * radius * radius * height;
    }
};

int main() {
```

```cpp
    double sphereRadius, cylinderRadius, cylinderHeight;

    cout << "Enter the radius of the sphere: ";
    cin >> sphereRadius;
```

```
5
4
6
```

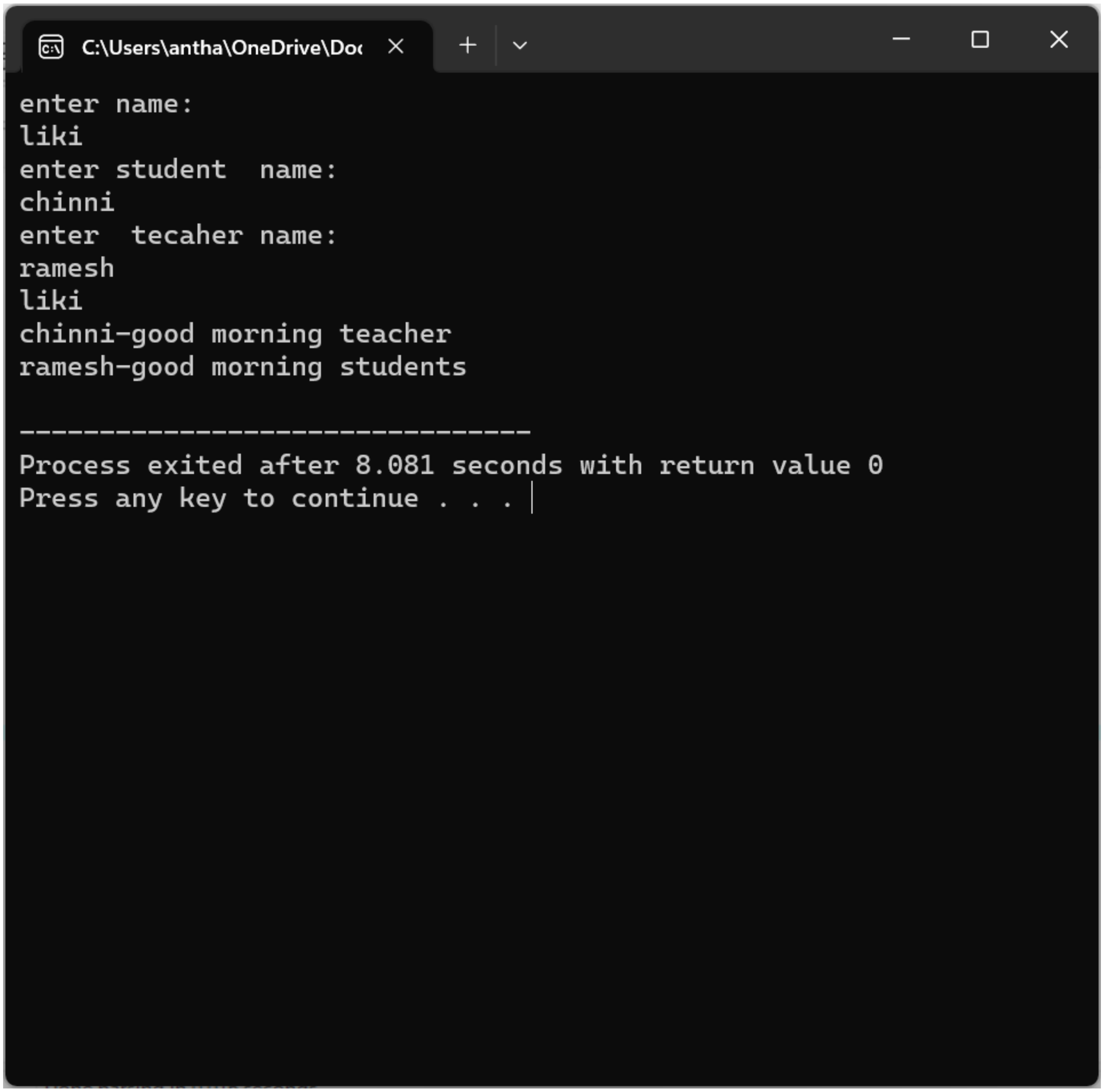
```
Enter the radius of the sphere:
Enter the radius of the cylinder:
Enter the height of the cylinder:

Sphere Area: 314.159
Sphere Volume: 523.599
Cylinder Area: 251.327
Cylinder Volume: 301.593
```

4)Create a base class called Person with a virtual function greet derive two classes Student and Teacher from the base class. Implement the greet function for each class


C:\Users\antha\OneDrive\Documents\Desktop\New folder\4.cpp - Dev-C++ 5.11
File
Edit
Search
View
Project
Execute
Tools
AStyle
Window
Help
TDM-GCC 4.9.2 64-bit Profiling
(globals)
Project
Classes
Debug
4.cpp

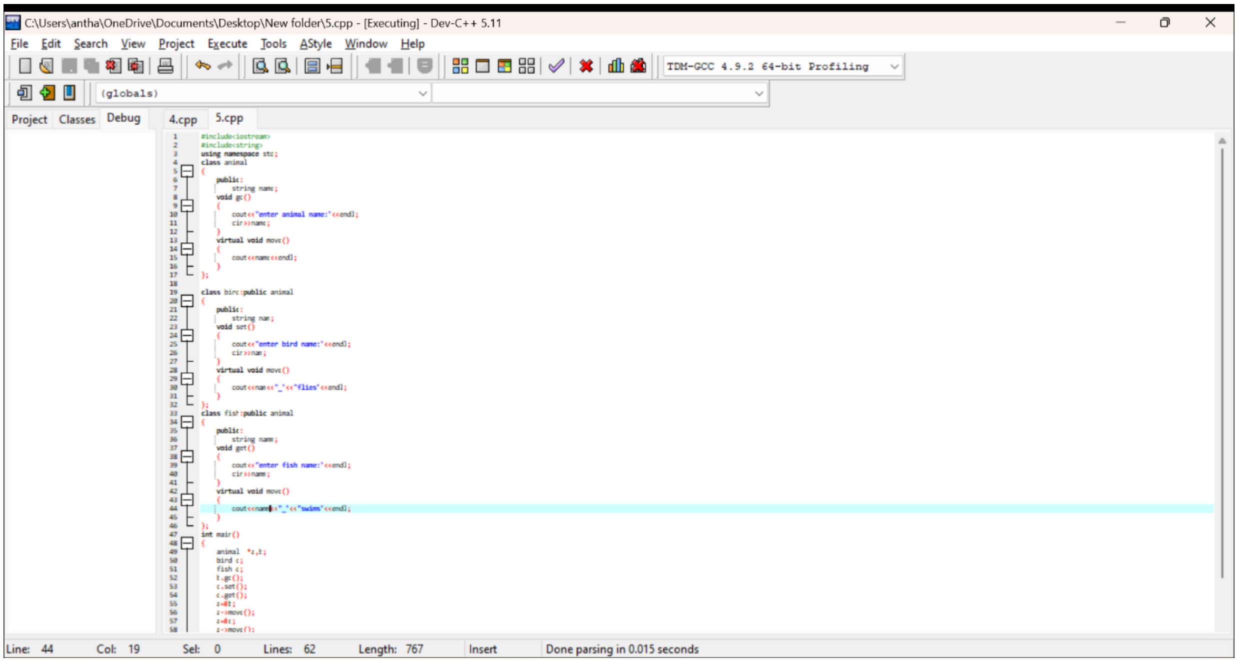
```
#include<iostream>
using namespace std;
class person
{
    public:
        string name;
    void gc()
    {
        cout<<"enter name:"<<endl;
        cir>>name;
    }
    virtual void greet()
    {
        cout<<name<<endl;
    }
};
class student: public person
{
    public:
        string nam;
    void set()
    {
        cout<<"enter name:"<<endl;
        cir>>nam;
    }
    void greet()
    {
        cout<<nam<<"-"<<"good morning teacher"<<endl;
    }
};
class teacher: public person
{
    public:
        string namm;
    void get()
    {
        cout<<"enter name:"<<endl;
        cir>>namm;
    }
    void greet()
    {
        cout<<namm<<"-"<<"good morning students"<<endl;
    }
};
int main()
{
    person *z,t;
    student s;
    teacher c;
    t.gc();
    s.set();
    c.get();
    z=&t;
    z->greet();
    z=&s;
    z->greet();
    z=&c;
    z->greet();
```

Line: 47 Col: 17 Sel: 0 Lines: 59 Length: 762 Insert Done parsing in 0.015 seconds



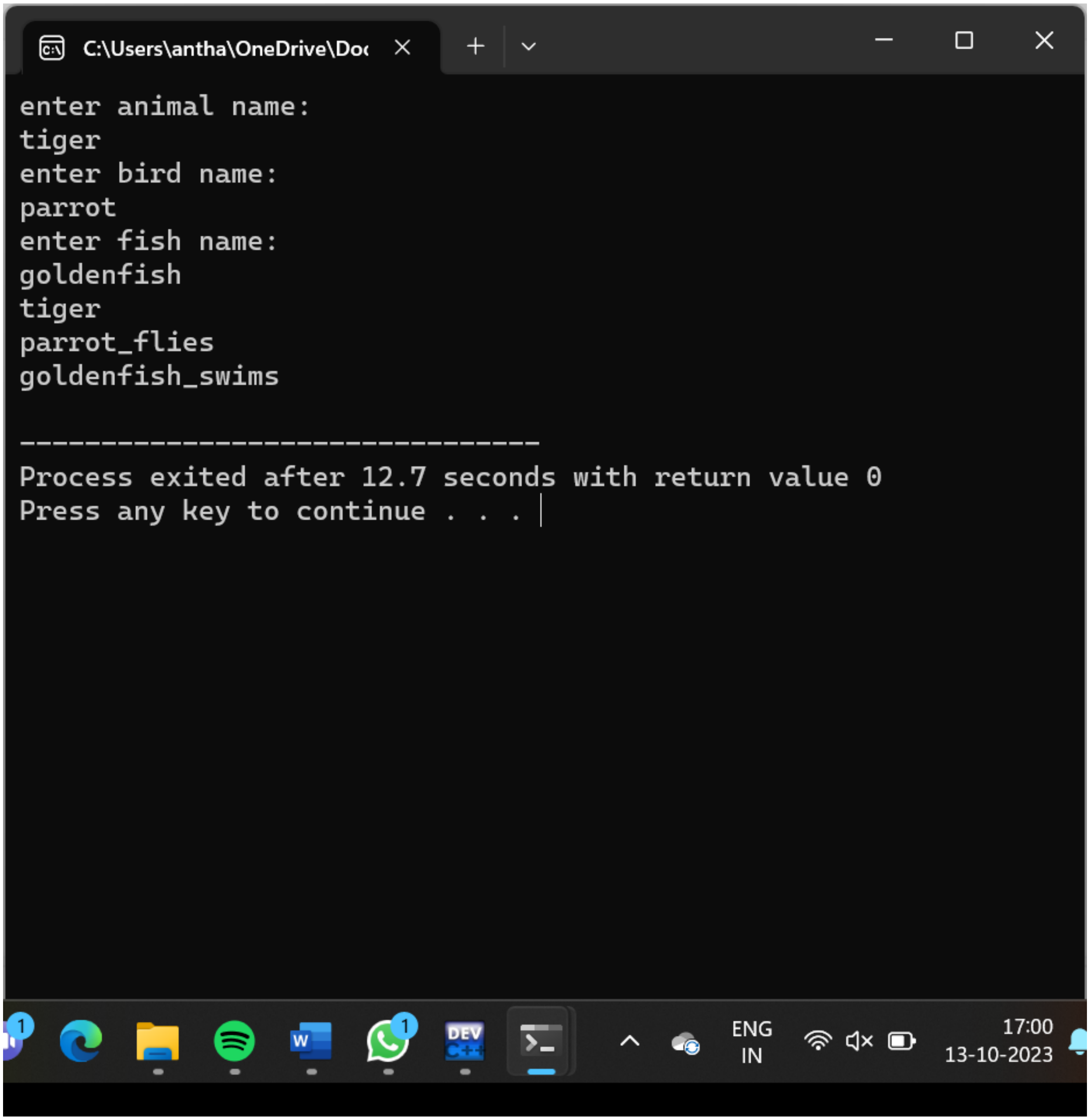
```
enter name:
liki
enter student  name:
chinni
enter  tecaher name:
ramesh
liki
chinni-good morning teacher
ramesh-good morning students

--------------------------------
Process exited after 8.081 seconds with return value 0
Press any key to continue . . .
```

5. Create a base class called Animal with a virtual function move derive two classes Bird and Fish from the base class. Implement the move function for each class.


C:\Users\antha\OneDrive\Documents\Desktop\New folder\5.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Profiling
(globals)
Project Classes Debug
4.cpp 5.cpp

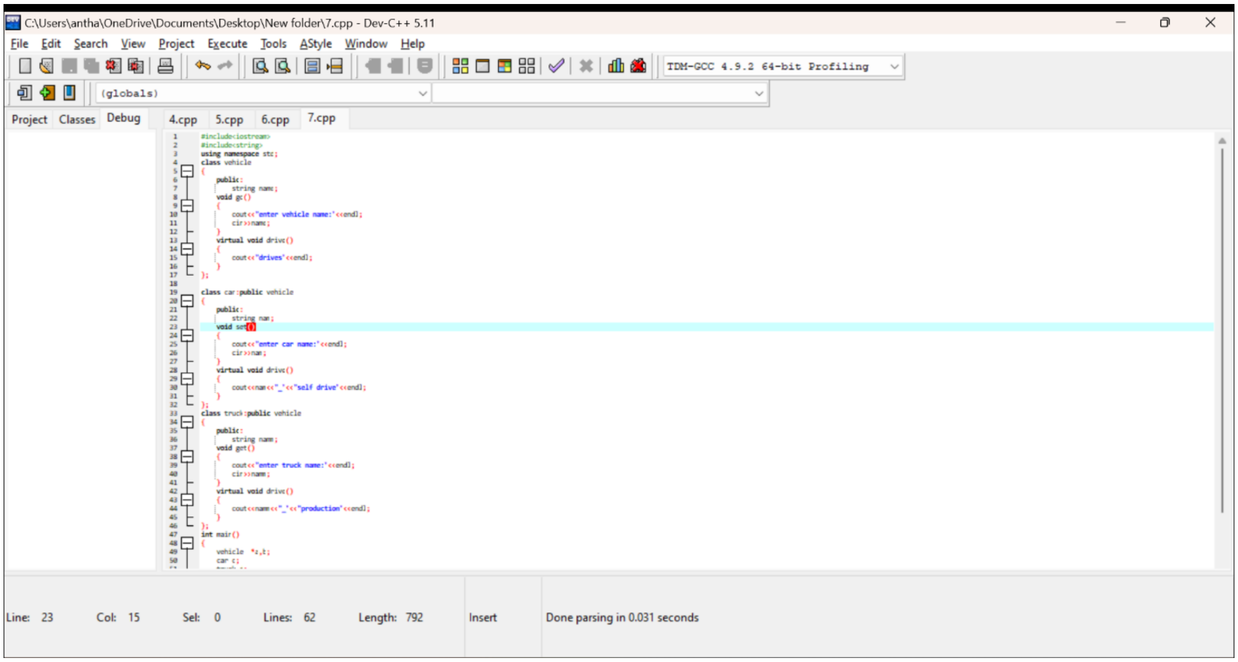
```cpp
#include<iostream>
#include<string>
using namespace std;
class animal
{
    public:
        string name;
    void ge()
    {
        cout<<"enter animal name:"<<endl;
        cin>>name;
    }
    virtual void move()
    {
        cout<<name<<endl;
    }
};

class bird:public animal
{
    public:
        string nam;
    void set()
    {
        cout<<"enter bird name:"<<endl;
        cin>>nam;
    }
    virtual void move()
    {
        cout<<nam<<"_"<<"flies"<<endl;
    }
};
class fish:public animal
{
    public:
        string namm;
    void get()
    {
        cout<<"enter fish name:"<<endl;
        cin>>namm;
    }
    virtual void move()
    {
        cout<<namm<<"_"<<"swims"<<endl;
    }
};
int main()
{
    animal  *z,t;
    bird c;
    fish e;
    t.ge();
    c.set();
    e.get();
    z=&t;
    z->move();
    z=&c;
    z->move();
```

Line: 44 Col: 19 Sel: 0 Lines: 62 Length: 767 Insert Done parsing in 0.015 seconds



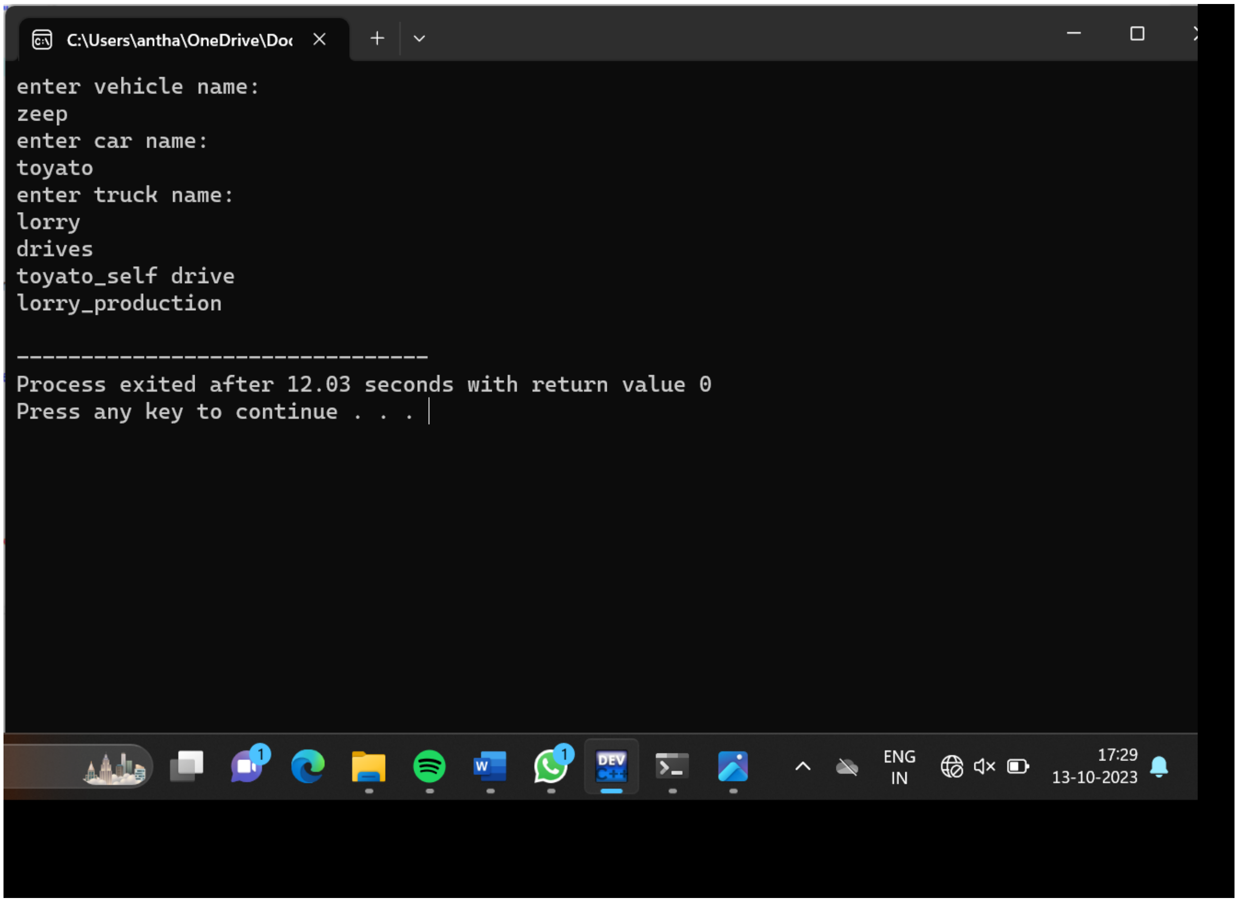
```
enter animal name:
tiger
enter bird name:
parrot
enter fish name:
goldenfish
tiger
parrot_flies
goldenfish_swims

--------------------------------
Process exited after 12.7 seconds with return value 0
Press any key to continue . . .
```

6..Create a base class called Vehicle with a virtual function drive derive two classes Car and Truck from the base class. Implement the drive function for each class


C:\Users\antha\OneDrive\Documents\Desktop\New folder\7.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Profiling
(globals)
Project Classes Debug
4.cpp 5.cpp 6.cpp 7.cpp

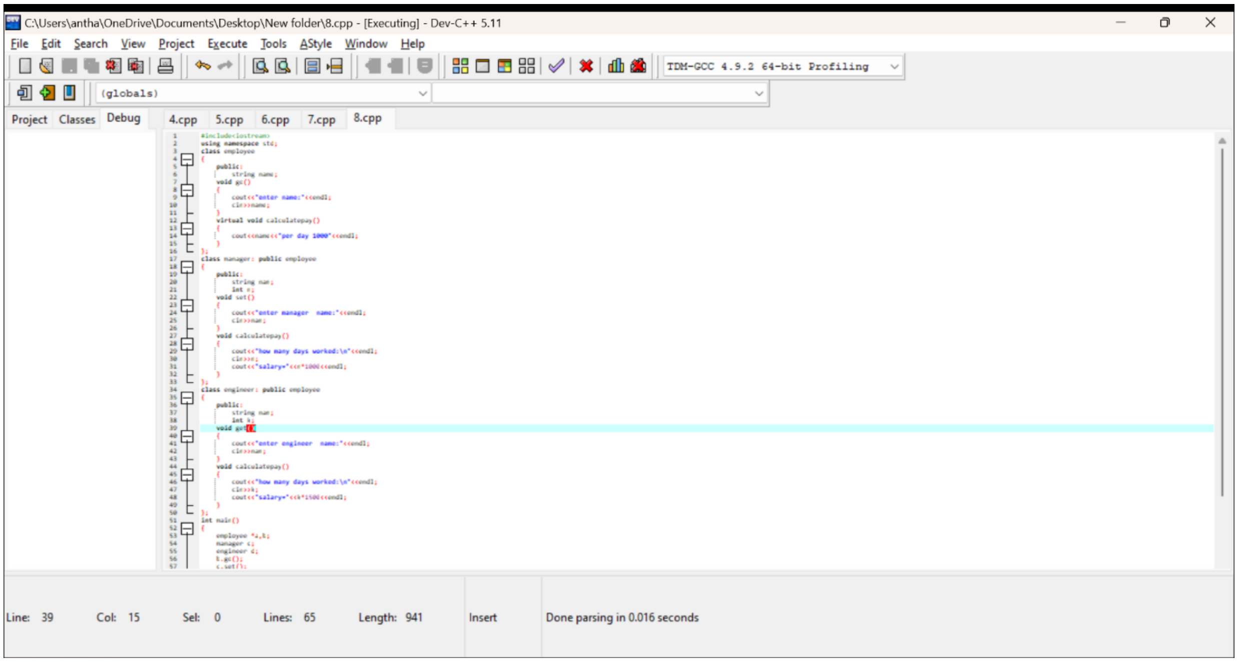
```cpp
#include<iostream>
#include<string>
using namespace std;
class vehicle
{
    public:
        string name;
    void get()
    {
        cout<<"enter vehicle name:"<<endl;
        cin>>name;
    }
    virtual void drive()
    {
        cout<<"drives"<<endl;
    }
};

class car:public vehicle
{
    public:
        string nam;
    void set()
    {
        cout<<"enter car name:"<<endl;
        cin>>nam;
    }
    virtual void drive()
    {
        cout<<nam<<"_"<<"self drive"<<endl;
    }
};
class truck:public vehicle
{
    public:
        string namm;
    void get()
    {
        cout<<"enter truck name:"<<endl;
        cin>>namm;
    }
    virtual void drive()
    {
        cout<<namm<<"_"<<"production"<<endl;
    }
};
int main()
{
    vehicle  *z,t;
    car c;
```

Line: 23 Col: 15 Sel: 0 Lines: 62 Length: 792 Insert Done parsing in 0.031 seconds



```
enter vehicle name:
zeep
enter car name:
toyato
enter truck name:
lorry
drives
toyato_self drive
lorry_production

--------------------------------
Process exited after 12.03 seconds with return value 0
Press any key to continue . . .
```

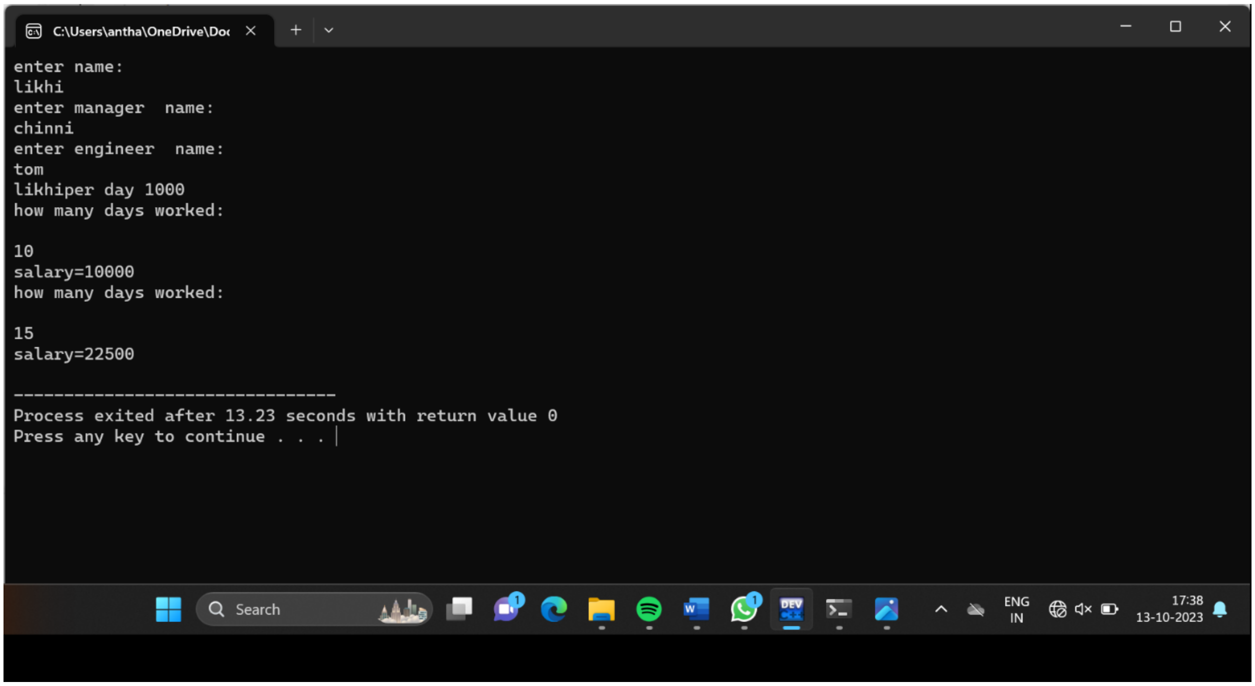
7)Create a base class called Employee with a virtual function calculatePay Derive two classes Manager and Engineer from the base class. Implement the calculatePay function for each class.



C:\Users\antha\OneDrive\Documents\Desktop\New folder\8.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Profiling
(globals)
Project Classes Debug
4.cpp 5.cpp 6.cpp 7.cpp 8.cpp

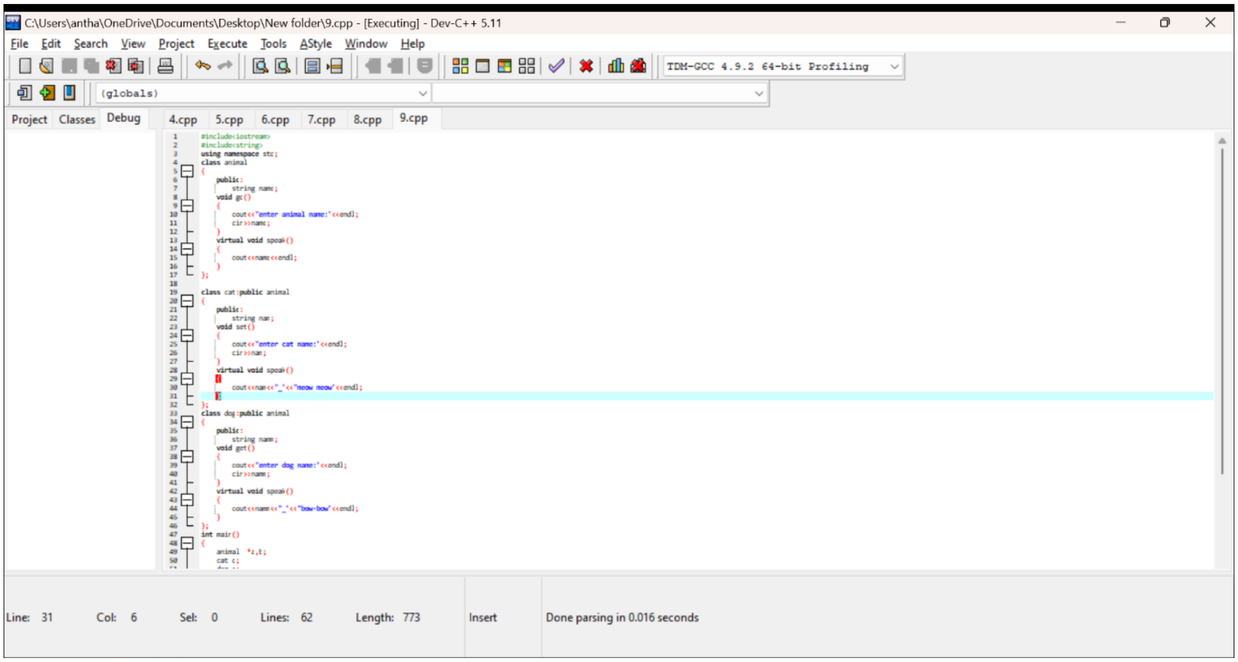
```
#include<iostream>
using namespace std;
class employee
{
    public:
        string name;
    void gc()
    {
        cout<<"enter name:"<<endl;
        cin>>name;
    }
    virtual void calculatepay()
    {
        cout<<name<<"per day 1000"<<endl;
    }
};
class manager: public employee
{
    public:
        string nan;
        int n;
    void set()
    {
        cout<<"enter manager  name:"<<endl;
        cin>>nan;
    }
    void calculatepay()
    {
        cout<<"how many days worked:\n"<<endl;
        cin>>n;
        cout<<"salary="<<n*1000<<endl;
    }
};
class engineer: public employee
{
    public:
        string nan;
        int k;
    void get()
    {
        cout<<"enter engineer  name:"<<endl;
        cin>>nan;
    }
    void calculatepay()
    {
        cout<<"how many days worked:\n"<<endl;
        cin>>k;
        cout<<"salary="<<k*1500<<endl;
    }
};
int main()
{
    employee *a,b;
    manager c;
    engineer d;
    b.gc();
    c.set();
```

Line: 39 Col: 15 Sel: 0 Lines: 65 Length: 941 Insert Done parsing in 0.016 seconds



```
enter name:
likhi
enter manager  name:
chinni
enter engineer  name:
tom
likhiper day 1000
how many days worked:

10
salary=10000
how many days worked:

15
salary=22500

--------------------------------
Process exited after 13.23 seconds with return value 0
Press any key to continue . . .
```

8)Create a base class called Animal with a virtual function speak Derive two classes Cat and Dog from the base class. Implement the speak function for each class.

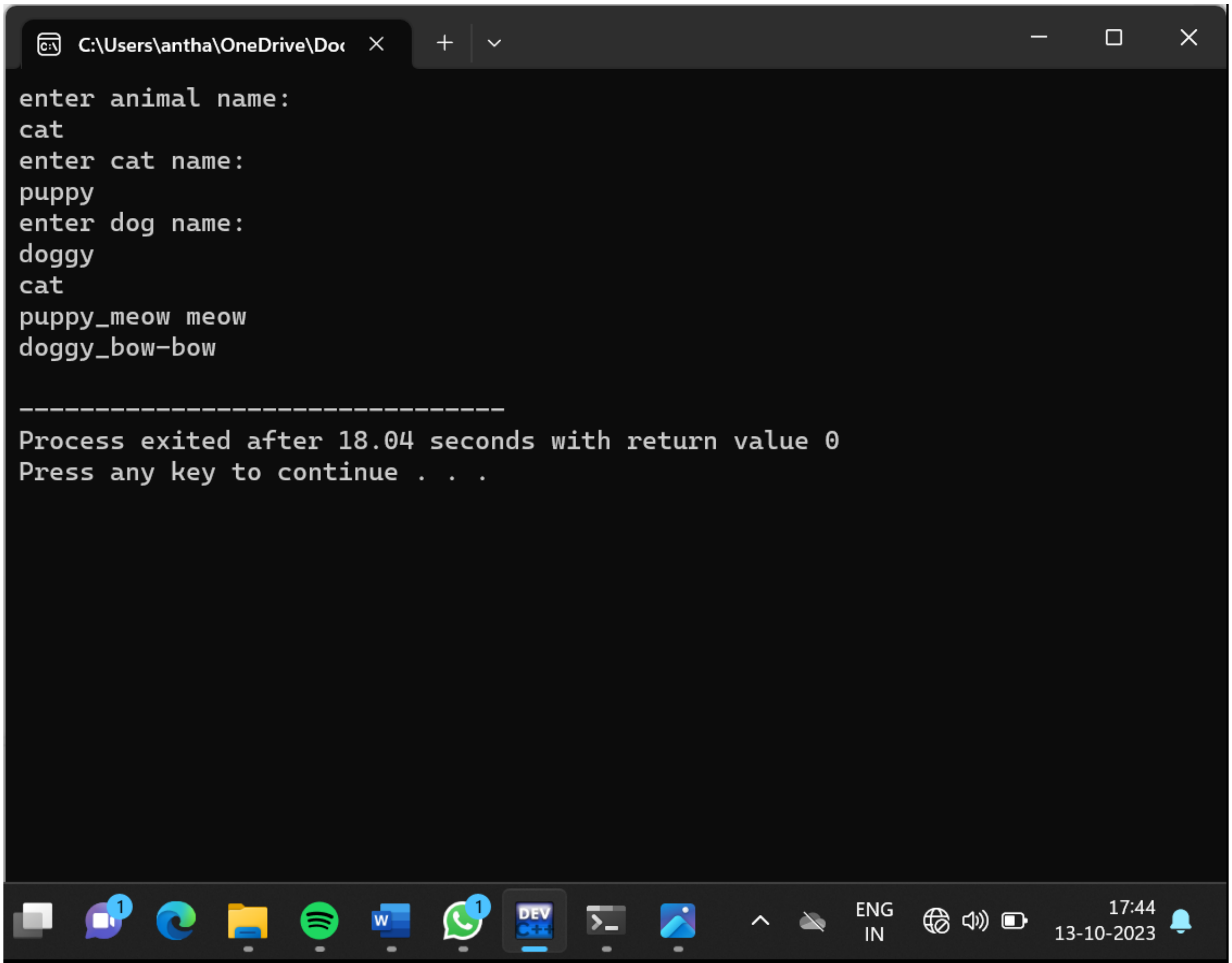


C:\Users\antha\OneDrive\Documents\Desktop\New folder\9.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Profiling
(globals)
Project Classes Debug
4.cpp 5.cpp 6.cpp 7.cpp 8.cpp 9.cpp

```
#include<iostream>
#include<string>
using namespace std;
class animal
{
	public:
		string name;
	void ge()
	{
		cout<<"enter animal name:"<<endl;
		cin>>name;
	}
	virtual void speak()
	{
		cout<<name<<endl;
	}
};

class cat:public animal
{
	public:
		string nam;
	void set()
	{
		cout<<"enter cat name:"<<endl;
		cin>>nam;
	}
	virtual void speak()
	{
		cout<<nam<<"_"<<"meow meow"<<endl;
	}
};
class dog:public animal
{
	public:
		string namm;
	void get()
	{
		cout<<"enter dog name:"<<endl;
		cin>>namm;
	}
	virtual void speak()
	{
		cout<<namm<<"_"<<"bow-bow"<<endl;
	}
};
int main()
{
	animal  *z,t;
	cat c;
```

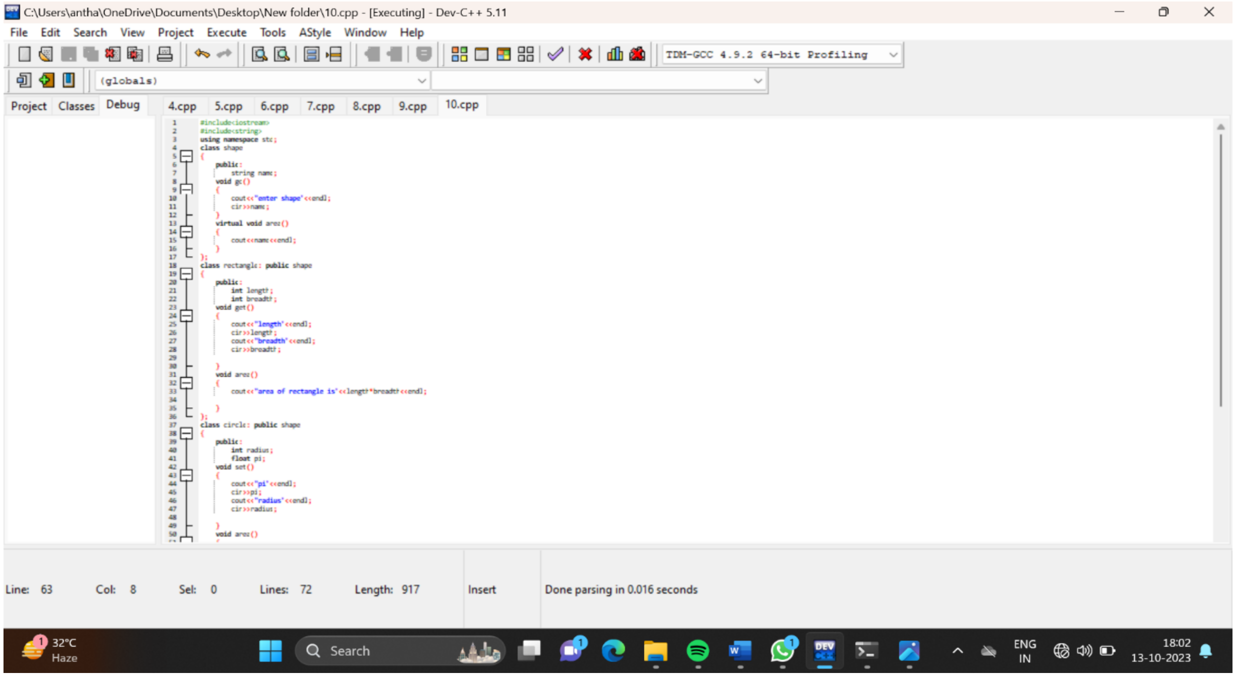
Line: 31 Col: 6 Sel: 0 Lines: 62 Length: 773 Insert Done parsing in 0.016 seconds



C:\Users\antha\OneDrive\Doc

```
enter animal name:
cat
enter cat name:
puppy
enter dog name:
doggy
cat
puppy_meow meow
doggy_bow-bow

--------------------------------
Process exited after 18.04 seconds with return value 0
Press any key to continue . . .
```

9)Create a base class called Shape with a virtual function area derive two classes Rectangle and Circle from the base class. Implement the area function for each class.

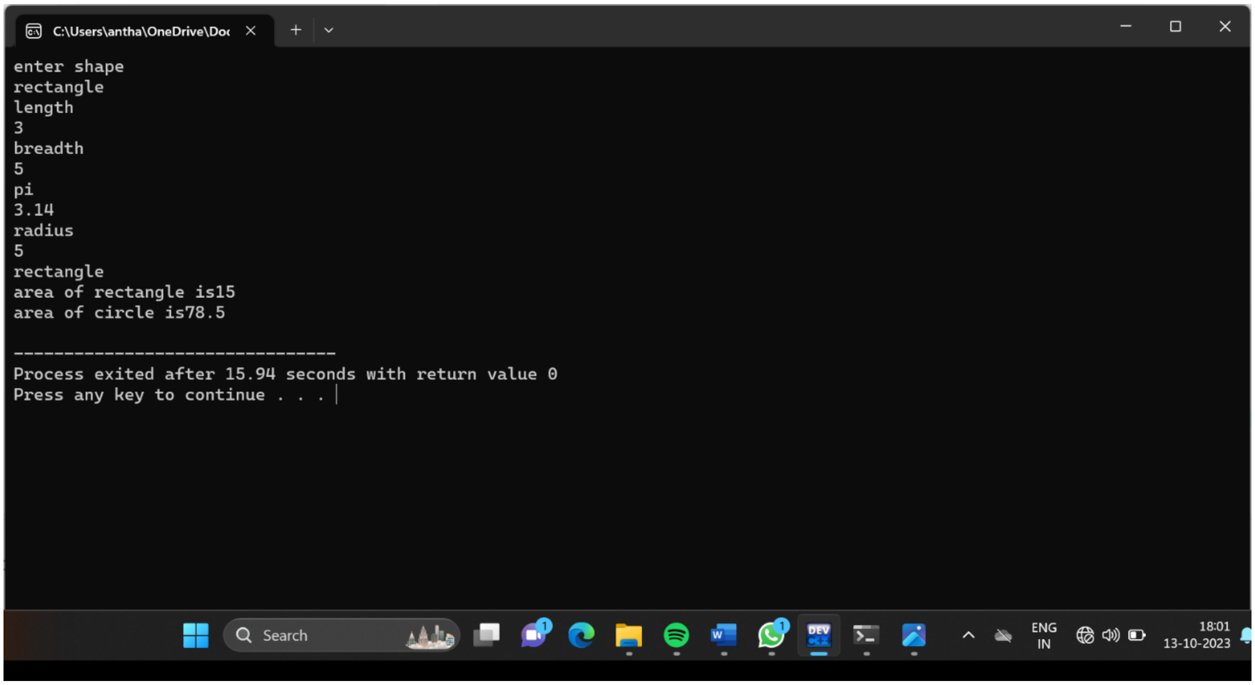


C:\Users\antha\OneDrive\Documents\Desktop\New folder\10.cpp - [Executing] - Dev-C++ 5.11

```
#include<iostream>
#include<string>
using namespace std;
class shape
{
    public:
        string name;
    void get()
    {
        cout<<"enter shape"<<endl;
        cin>>name;
    }
    virtual void area()
    {
        cout<<name<<endl;
    }
};
class rectangle: public shape
{
    public:
        int length;
        int breadth;
    void get()
    {
        cout<<"length"<<endl;
        cin>>length;
        cout<<"breadth"<<endl;
        cin>>breadth;

    }
    void area()
    {
        cout<<"area of rectangle is"<<length*breadth<<endl;

    }
};
class circle: public shape
{
    public:
        int radius;
        float pi;
    void set()
    {
        cout<<"pi"<<endl;
        cin>>pi;
        cout<<"radius"<<endl;
        cin>>radius;

    }
    void area()
```

Line: 63 Col: 8 Sel: 0 Lines: 72 Length: 917 Insert Done parsing in 0.016 seconds

File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Profiling
(globals)
Project Classes Debug
4.cpp 5.cpp 6.cpp 7.cpp 8.cpp 9.cpp 10.cpp
32°C Haze
Search
ENG IN
18:02 13-10-2023

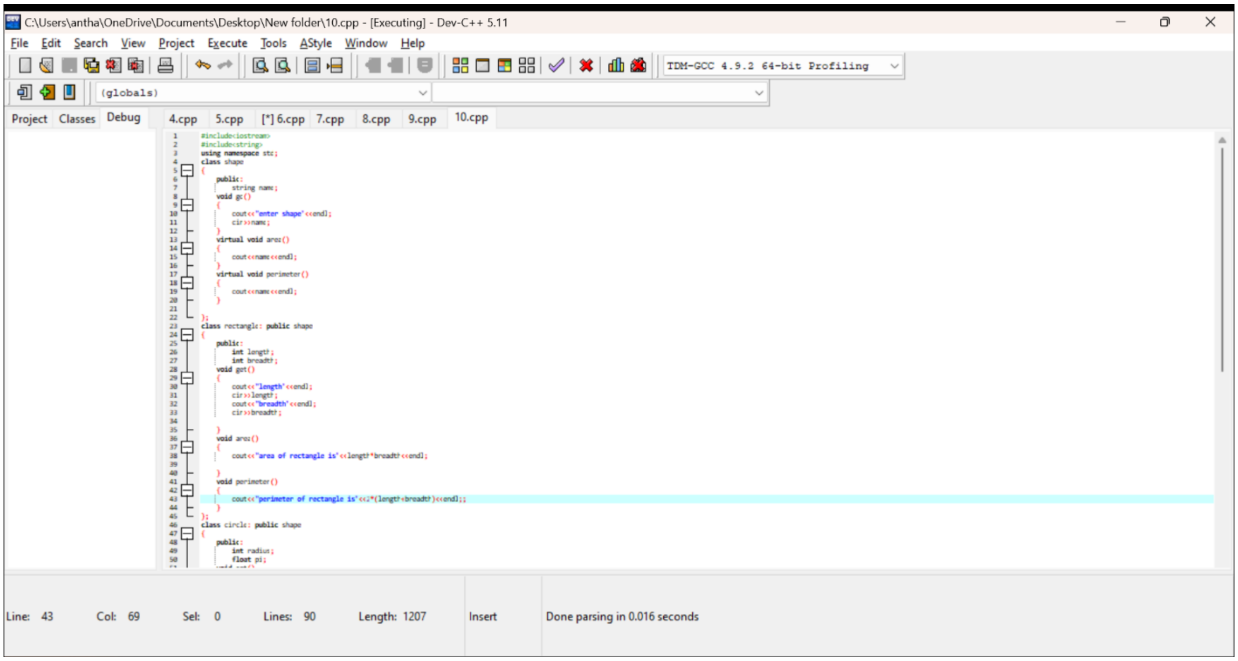


C:\Users\antha\OneDrive\Doc

```
enter shape
rectangle
length
3
breadth
5
pi
3.14
radius
5
rectangle
area of rectangle is15
area of circle is78.5

--------------------------------
Process exited after 15.94 seconds with return value 0
Press any key to continue . . .
```

Search
ENG IN
18:01 13-10-2023

10) Create a base class called Shape with virtual functions area and perimeter Derive two classes Rectangle and Triangle from the base class. Implement the area and perimeter functions for each class

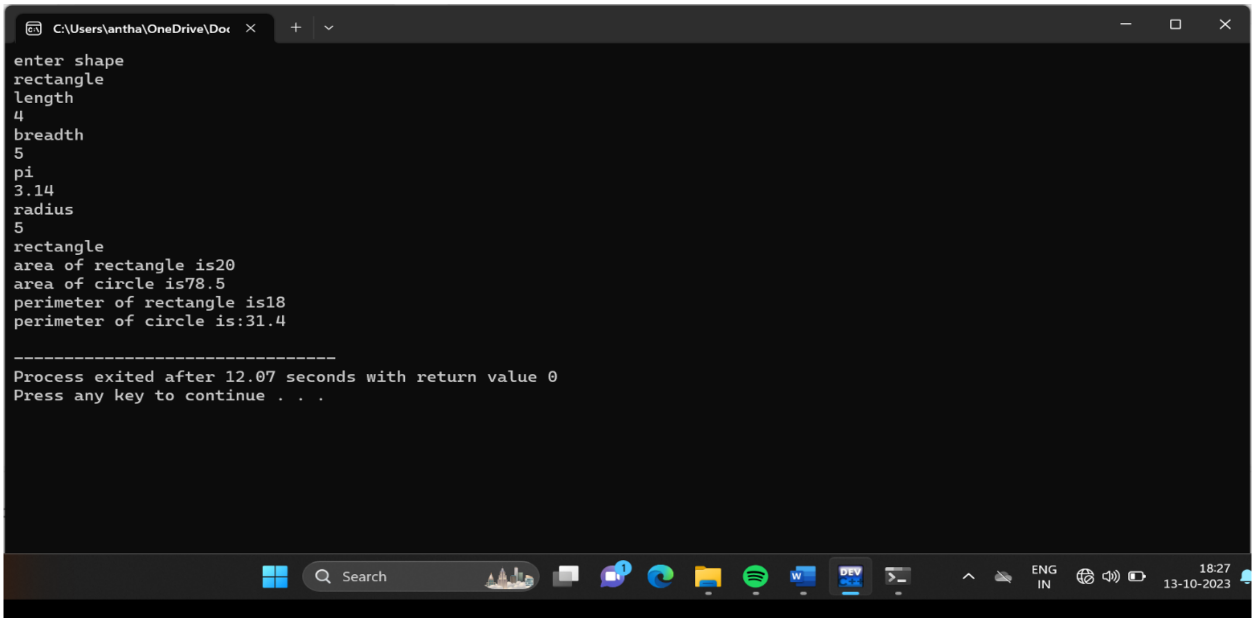


C:\Users\antha\OneDrive\Documents\Desktop\New folder\10.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Profiling
(globals)
Project Classes Debug
4.cpp 5.cpp [*] 6.cpp 7.cpp 8.cpp 9.cpp 10.cpp

```cpp
#include<iostream>
#include<string>
using namespace std;
class shape
{
	public:
		string name;
	void get()
	{
		cout<<"enter shape"<<endl;
		cin>>name;
	}
	virtual void area()
	{
		cout<<name<<endl;
	}
	virtual void perimeter()
	{
		cout<<name<<endl;
	}

};
class rectangle: public shape
{
	public:
		int length;
		int breadth;
	void get()
	{
		cout<<"length"<<endl;
		cin>>length;
		cout<<"breadth"<<endl;
		cin>>breadth;

	}
	void area()
	{
		cout<<"area of rectangle is"<<length*breadth<<endl;

	}
	void perimeter()
	{
		cout<<"perimeter of rectangle is"<<2*(length+breadth)<<endl;;
	}
};
class circle: public shape
{
	public:
		int radius;
		float pi;
	void get()
```

Line: 43 Col: 69 Sel: 0 Lines: 90 Length: 1207 Insert Done parsing in 0.016 seconds



C:\Users\antha\OneDrive\Doc

```
enter shape
rectangle
length
4
breadth
5
pi
3.14
radius
5
rectangle
area of rectangle is20
area of circle is78.5
perimeter of rectangle is18
perimeter of circle is:31.4

--------------------------------
Process exited after 12.07 seconds with return value 0
Press any key to continue . . .
```