

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";
    }
};

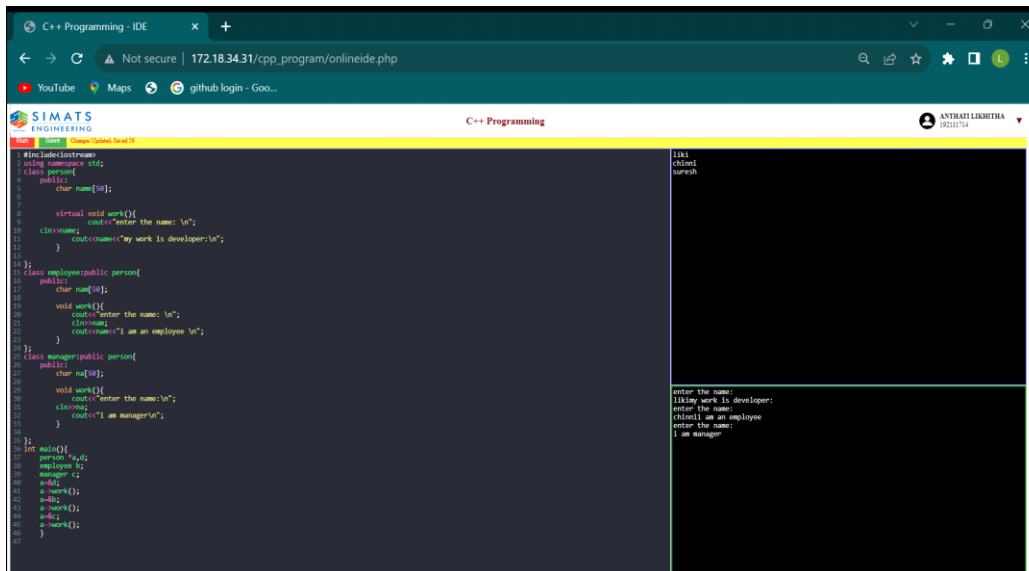
class employee:public person{
public:
    char nam[50];

    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();
}
```



The screenshot shows a C++ IDE window titled "C++ Programming - IDE". The browser address bar indicates the URL "172.18.34.31/cpp_program/onlineide.php". The code editor contains the same C++ code as the previous block. The output window on the right shows the following text:

```
liki
chimi
surah
enter the name:
likikay work is developer:
enter the name:
chimi i 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.

```

1 #include<iostream>
2 using namespace std;
3 class animal
4 {
5     public:
6     string name;
7     void go()
8     {
9         cout<<"enter animal name:"<<endl;
10        cin>>name;
11    }
12    virtual void eat()
13    {
14        cout<<name<<endl;
15    }
16 };
17 class herbivore: public animal
18 {
19     public:
20     string nam;
21     void set()
22     {
23         cout<<"enter herbivore:"<<endl;
24         cin>>nam;
25     }
26     void eat()
27     {
28         cout<<nam<<"_"<<"eats only veg"<<endl;
29     }
30 };
31 class carnivore: public animal
32 {
33     public:
34     string namn;
35     void get()
36     {
37         cout<<"enter carnivore:"<<endl;
38         cin>>namn;
39     }
40     void eat()
41     {
42         cout<<namn<<"_"<<"eats both veg and non veg"<<endl;
43     }
44 }

```

The screenshot shows a web-based C++ IDE. The left pane contains the C++ code, and the right pane shows the output of the program. The output demonstrates the execution of the program with inputs 'lion' and 'tiger'.

```

1 #include<iostream>
2 using namespace std;
3 class animal
4 {
5     public:
6     string name;
7     void go()
8     {
9         cout<<"enter animal name:"<<endl;
10        cin>>name;
11    }
12    virtual void eat()
13    {
14        cout<<name<<endl;
15    }
16 };
17 class herbivore: public animal
18 {
19     public:
20     string nam;
21     void set()
22     {
23         cout<<"enter herbivore:"<<endl;
24         cin>>nam;
25     }
26     void eat()
27     {
28         cout<<nam<<"_"<<"eats only veg"<<endl;
29     }
30 };
31 class carnivore: public animal
32 {
33     public:
34     string namn;
35     void get()
36     {
37         cout<<"enter carnivore:"<<endl;
38         cin>>namn;
39     }
40     void eat()
41     {
42         cout<<namn<<"_"<<"eats both veg and non veg"<<endl;
43     }
44 }
45
46 int main()
47 {
48     animal *a;
49     herbivore h;
50     carnivore c;
51 }

```

Output:

```

lion
tiger

enter animal name:
enter herbivore:
enter carnivore:
lion
tiger_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.

```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 class Shape {
6 public:
7     virtual double area() {
8         return 0.0;
9     }
10
11     virtual double volume() {
12         return 0.0;
13     }
14 };
15
16 class Sphere : public Shape {
17 public:
18     double radius;
19     Sphere(double r) : radius(r) {}
20
21     double area() {
22         return 4 * M_PI * radius * radius;
23     }
24
25     double volume() {
26         return (4.0 / 3.0) * M_PI * radius * radius * radius;
27     }
28 };
29
30 class Cylinder : public Shape {
31 public:
32     double radius;
33     double height;
34     Cylinder(double r, double h) : radius(r), height(h) {}
35
36     double area() {
37         return 2 * M_PI * radius * (radius + height);
38     }
39
40     double volume() {
41         return M_PI * radius * radius * height;
42     }
43 };
44
45 int main() {

```

```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 class Shape {
6 public:
7     virtual double area() {
8         return 0.0;
9     }
10
11     virtual double volume() {
12         return 0.0;
13     }
14 };
15
16 class Sphere : public Shape {
17 public:
18     double radius;
19     Sphere(double r) : radius(r) {}
20
21     double area() {
22         return 4 * M_PI * radius * radius;
23     }
24
25     double volume() {
26         return (4.0 / 3.0) * M_PI * radius * radius * radius;
27     }
28 };
29
30 class Cylinder : public Shape {
31 public:
32     double radius;
33     double height;
34     Cylinder(double r, double h) : radius(r), height(h) {}
35
36     double area() {
37         return 2 * M_PI * radius * (radius + height);
38     }
39
40     double volume() {
41         return M_PI * radius * radius * height;
42     }
43 };
44
45 int main() {
    double sphereradius, cylinderradius, cylindereheight;
    cout << "Enter the radius of the sphere: ";
    cin >> sphereradius;

```

```

Enter the radius of the sphere:
Enter the radius of the cylinder:
Enter the height of the cylinder:
Sphere Area: 344.559
Sphere Volume: 523.599
Cylinder Area: 251.327
Cylinder Volume: 361.583

```

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

TCM-GCC 4.9.2 64-bit Profiling

(globals)

Project Classes Debug 4.cpp

```
1 #include <iostream>
2 using namespace std;
3 class person
4 {
5     public:
6     string name;
7     void g()
8     {
9         cout<<"enter name"<<endl;
10        cin>>name;
11    }
12    virtual void greet()
13    {
14        cout<<name<<endl;
15    }
16 }
17 class student: public person
18 {
19     public:
20     string nam;
21     void set()
22     {
23         cout<<"enter name"<<endl;
24         cin>>nam;
25     }
26     void greet()
27     {
28         cout<<nam<<"<<"good morning teacher"<<endl;
29     }
30 }
31 class teacher: public person
32 {
33     public:
34     string name;
35     void get()
36     {
37         cout<<"enter name"<<endl;
38         cin>>name;
39     }
40     void greet()
41     {
42         cout<<name<<"<<"good morning students"<<endl;
43     }
44 }
45 int main()
46 {
47     person *p;
48     student s;
49     teacher t;
50     t.g();
51     s.set();
52     t.greet();
53     s.greet();
54     t.greet();
55     s.greet();
56     t.greet();
57 }
58
```

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

```
C:\Users\antha\OneDrive\Doc  X + v - □ X
enter name:
liki
enter student name:
chinni
enter teacher 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

TCM-GCC 4.9.2 64-bit Profiling

(globals)


Project Classes Debug 4.cpp 5.cpp

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class animal
5 {
6     public:
7     string name;
8     void go()
9     {
10         cout<<"enter animal name"<<endl;
11         cin>>name;
12     }
13     virtual void move()
14     {
15         cout<<name<<endl;
16     }
17 };
18 class bird:public animal
19 {
20     public:
21     string name;
22     void set()
23     {
24         cout<<"enter bird name"<<endl;
25         cin>>name;
26     }
27     virtual void move()
28     {
29         cout<<name<<"_"<<"flies"<<endl;
30     }
31 };
32 class fish:public animal
33 {
34     public:
35     string name;
36     void get()
37     {
38         cout<<"enter fish name"<<endl;
39         cin>>name;
40     }
41     virtual void move()
42     {
43         cout<<name<<"_"<<"swims"<<endl;
44     }
45 };
46 int main()
47 {
48     animal *a;
49     bird b;
50     fish c;
51     a=&b;
52     a->get();
53     a->move();
54     a=&c;
55     a->move();
56     a->move();
57 }
```

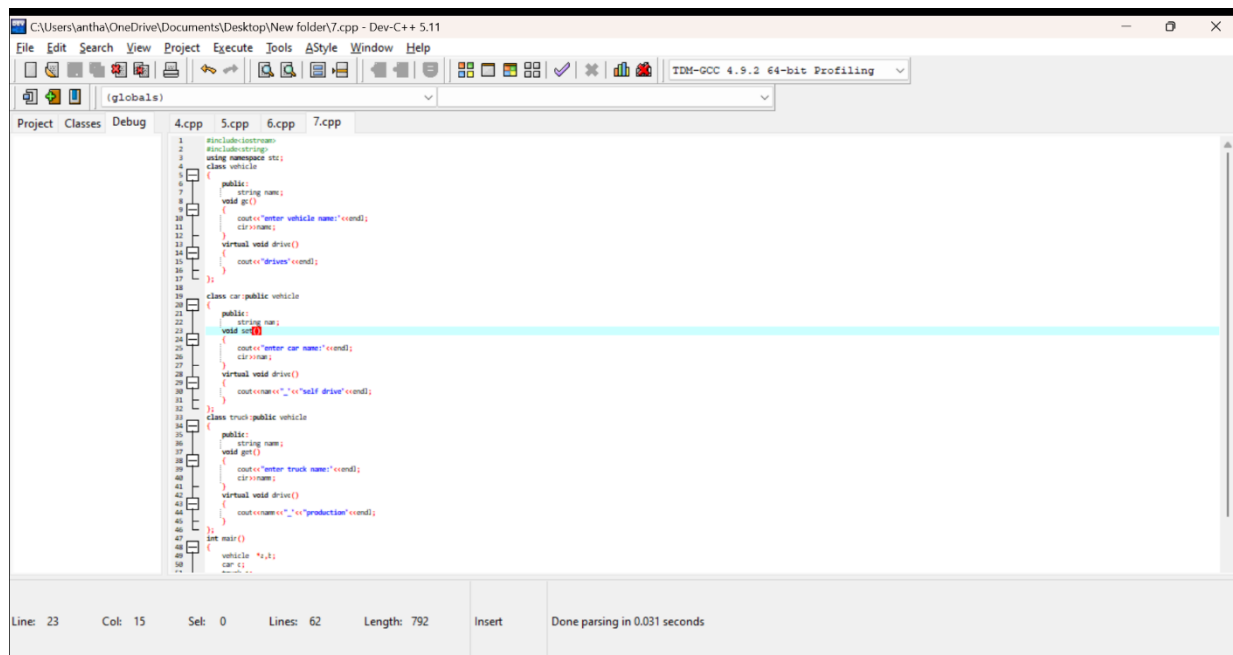
Line: 44 Col: 19 Sel: 0 Lines: 62 Length: 767 Insert Done parsing in 0.015 seconds

```
C:\Users\antha\OneDrive\Doc  X + v - □ X
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 . . . |
```



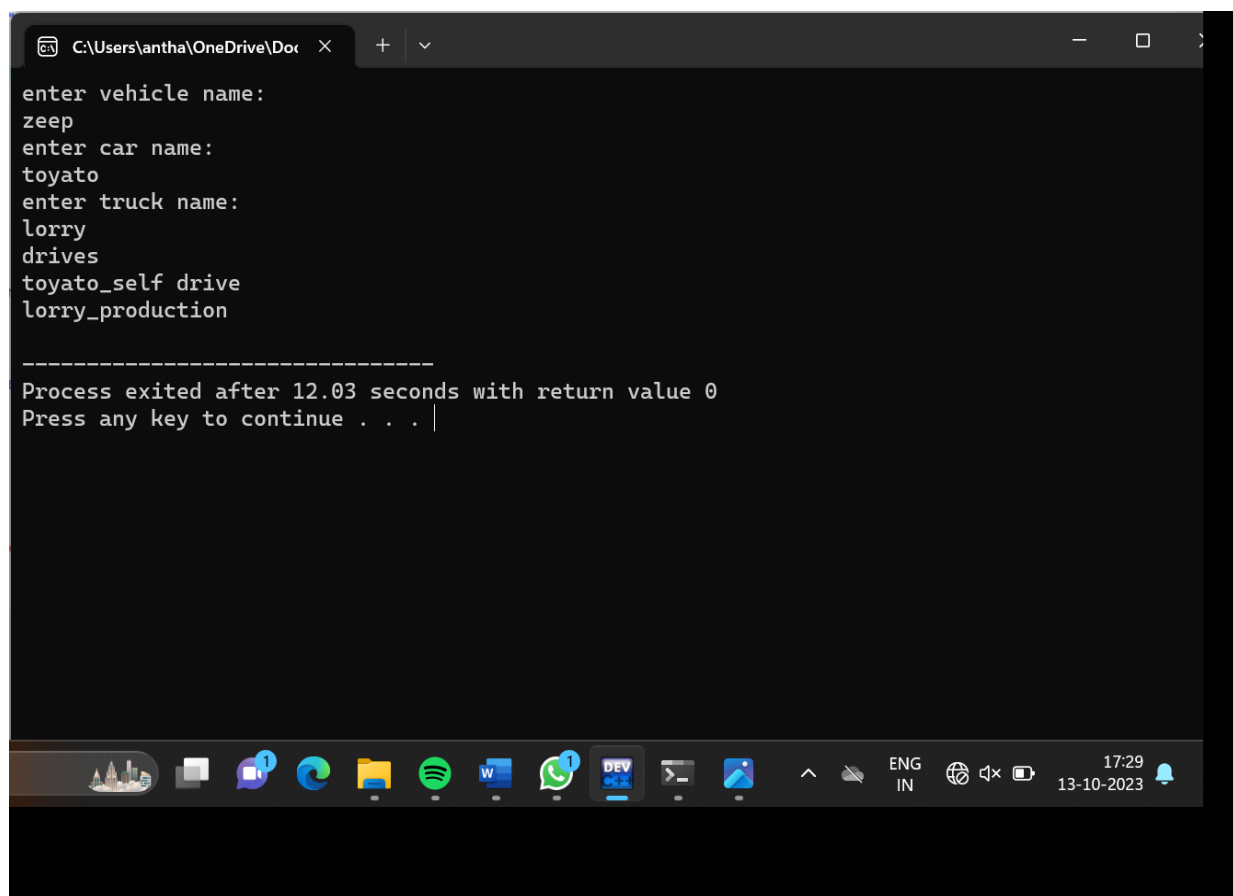
6..Create a base class called Vehicle with a virtual function drive derive two classes Car and Truck from the base class. Implement the drive function for each class



The screenshot shows the Dev-C++ IDE with a C++ program. The code defines a base class `vehicle` and two derived classes, `car` and `truck`. The `vehicle` class has a `name` attribute, a `get()` method, and a virtual `drive()` method. The `car` class overrides `drive()` to print "toyato_self drive", and the `truck` class overrides it to print "lorry_production". The `main()` function creates a `vehicle` object and calls its `get()` and `drive()` methods.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class vehicle
5 {
6     public:
7     string name;
8     void get()
9     {
10         cout<<"enter vehicle name:"<<endl;
11         cin>>name;
12     }
13     virtual void drive()
14     {
15         cout<<"drives"<<endl;
16     }
17 };
18 class car:public vehicle
19 {
20     public:
21     string name;
22     void get()
23     {
24         cout<<"enter car name:"<<endl;
25         cin>>name;
26     }
27     virtual void drive()
28     {
29         cout<<name<<"_self drive"<<endl;
30     }
31 };
32 class truck:public vehicle
33 {
34     public:
35     string name;
36     void get()
37     {
38         cout<<"enter truck name:"<<endl;
39         cin>>name;
40     }
41     virtual void drive()
42     {
43         cout<<name<<"_production"<<endl;
44     }
45 };
46 int main()
47 {
48     vehicle *v;
49     car c;
50     truck t;
51 }
```

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

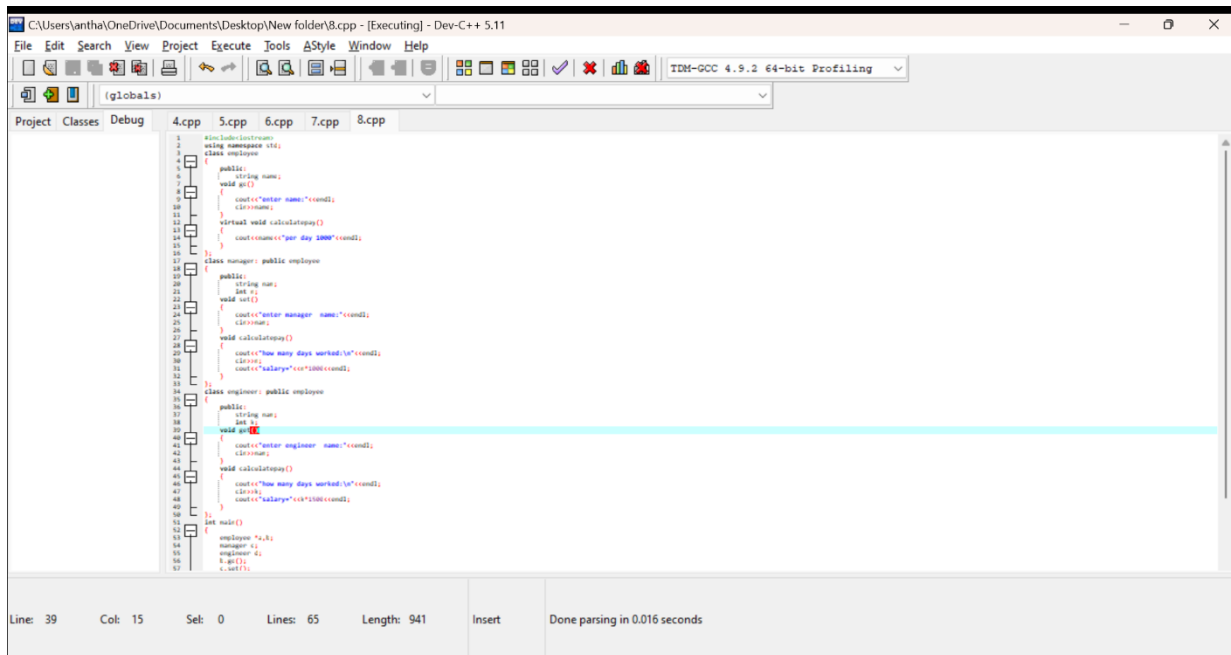


The screenshot shows a terminal window with the output of the C++ program. The program prompts the user to enter a vehicle name, a car name, and a truck name. The user enters "zeep", "toyato", and "lorry" respectively. The program then prints "drives", "toyato_self drive", and "lorry_production". The terminal also shows the process exit message and a prompt to press any key to continue.

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

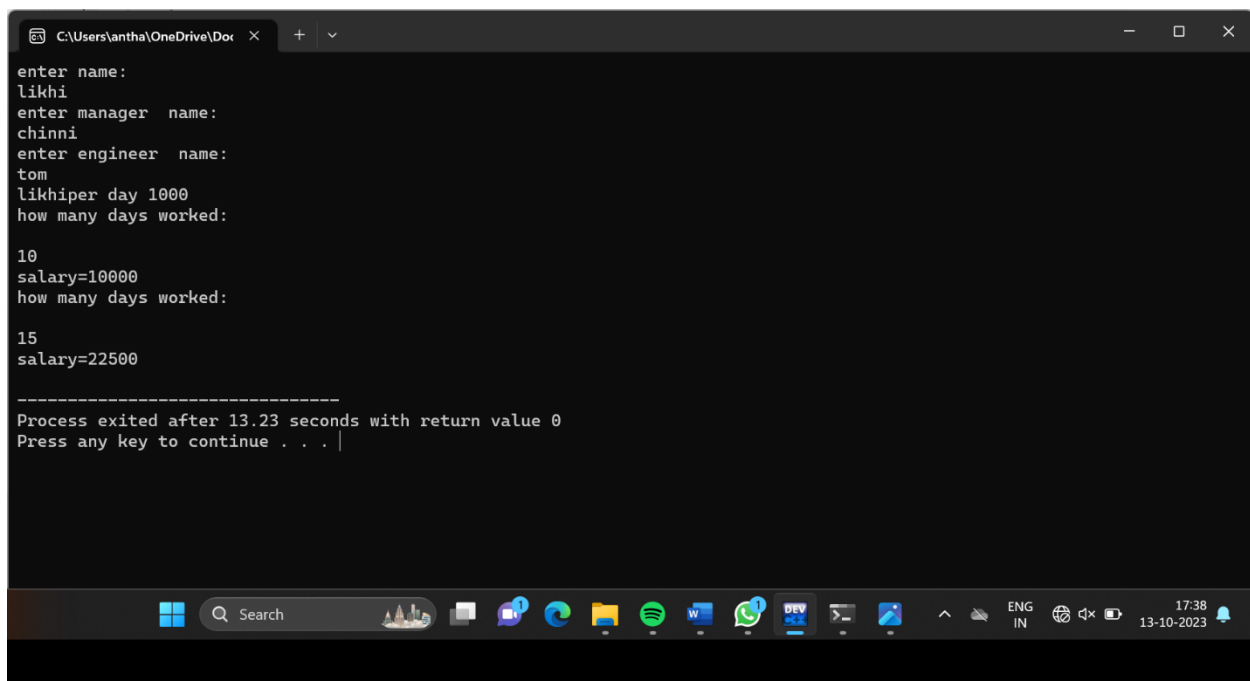
-----
Process exited after 12.03 seconds with return value 0
Press any key to continue . . .
```

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.



```
1 #include<iostream>
2 using namespace std;
3 class employee
4 {
5     public:
6         string name;
7         void get()
8         {
9             cout<<"enter name:"<<endl;
10             cin>>name;
11         }
12         virtual void calculatepay()
13         {
14             cout<<"per day 1000"<<endl;
15         }
16 };
17 class manager: public employee
18 {
19     public:
20         string name;
21         int i;
22         void set()
23         {
24             cout<<"enter manager name:"<<endl;
25             cin>>name;
26         }
27         void calculatepay()
28         {
29             cout<<"how many days worked:"<<endl;
30             cin>>i;
31             cout<<"salary"<<i*1000<<endl;
32         }
33 };
34 class engineer: public employee
35 {
36     public:
37         string name;
38         int i;
39         void get()
40         {
41             cout<<"enter engineer name:"<<endl;
42             cin>>name;
43         }
44         void calculatepay()
45         {
46             cout<<"how many days worked:"<<endl;
47             cin>>i;
48             cout<<"salary"<<i*800<<endl;
49         }
50 };
51 int main()
52 {
53     employee *e;
54     manager m;
55     engineer d;
56     e.get();
57     e.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
(globals)
Project Classes Debug 4.cpp 5.cpp 6.cpp 7.cpp 8.cpp 9.cpp
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class animal
5 {
6 public:
7     string name;
8     void get()
9     {
10         cout << "enter animal name:" << endl;
11         cin >> name;
12     }
13     virtual void speak()
14     {
15         cout << name << endl;
16     }
17 };
18 class cat: public animal
19 {
20 public:
21     string name;
22     void set()
23     {
24         cout << "enter cat name:" << endl;
25         cin >> name;
26     }
27     virtual void speak()
28     {
29         cout << name << " " << "meow meow" << endl;
30     }
31 };
32 class dog: public animal
33 {
34 public:
35     string name;
36     void get()
37     {
38         cout << "enter dog name:" << endl;
39         cin >> name;
40     }
41     virtual void speak()
42     {
43         cout << name << " " << "bow-bow" << endl;
44     }
45 };
46 int main()
47 {
48     animal *a;
49     cat c1;
50     dog d1;
51 }
```

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

```
C:\Users\antha\OneDrive\Documents\Desktop\New folder\9.cpp - [Executing] - Dev-C++ 5.11
enter animal name:
cat
enter cat name:
puppy
enter dog name:
doggy
cat
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.

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class shape
5 {
6 public:
7     string name;
8     void get()
9     {
10         cout << "enter shape" << endl;
11         cin >> name;
12     }
13     virtual void area()
14     {
15         cout << name << endl;
16     }
17 };
18 class rectangle: public shape
19 {
20 public:
21     int length;
22     int breadth;
23     void get()
24     {
25         cout << "length" << endl;
26         cin >> length;
27         cout << "breadth" << endl;
28         cin >> breadth;
29     }
30     void area()
31     {
32         cout << "area of rectangle is" << length * breadth << endl;
33     }
34 };
35 class circle: public shape
36 {
37 public:
38     float radius;
39     void set()
40     {
41         cout << "pi" << endl;
42         cin >> pi;
43         cout << "radius" << endl;
44         cin >> radius;
45     }
46     void area()
47     {
48         // Implementation of area for circle
49     }
50 };

```

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

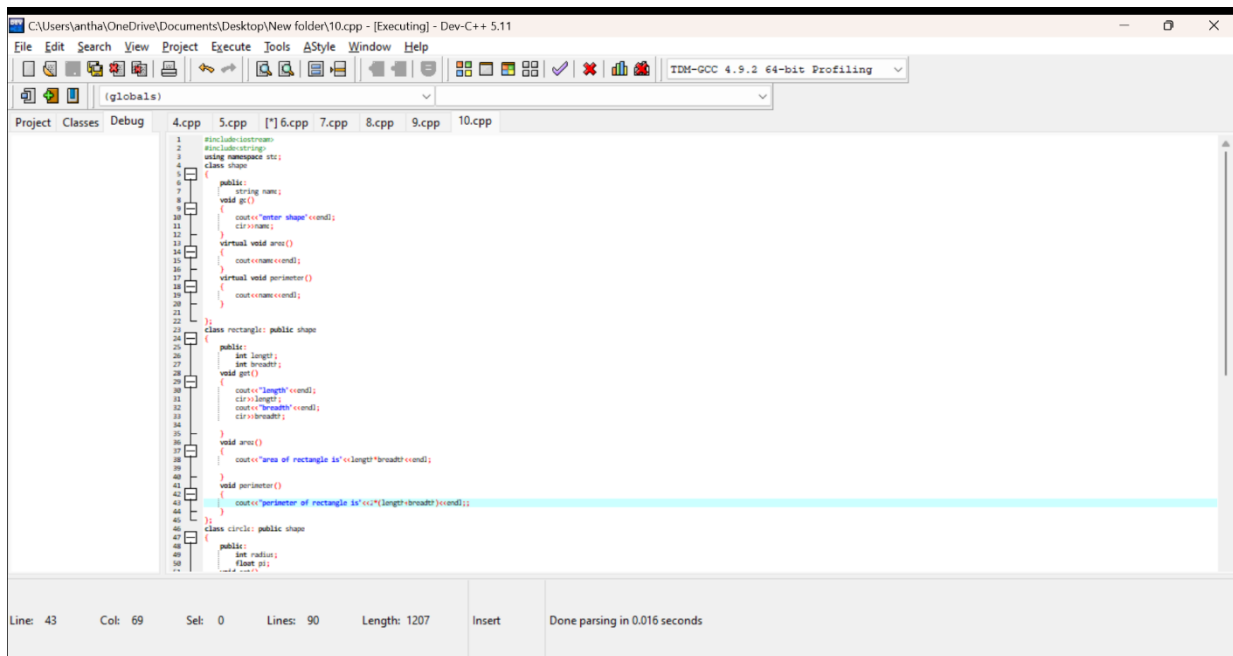
```

C:\Users\antha\OneDrive\Documents\Desktop\New folder\10.cpp - [Executing] - Dev-C++ 5.11
enter shape
rectangle
length
3
breadth
5
pi
3.14
radius
5
rectangle
area of rectangle is 15
area of circle is 78.5

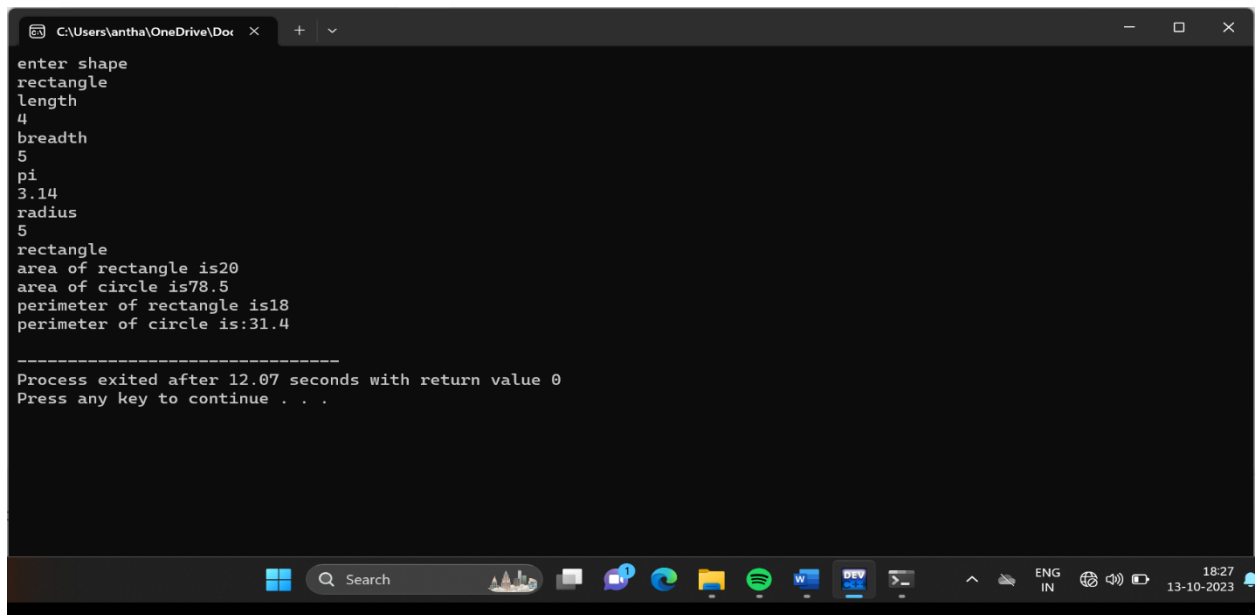
-----
Process exited after 15.94 seconds with return value 0
Press any key to continue . . .

```

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



```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class shape
5 {
6 public:
7     string name;
8     void get()
9     {
10         cout << "enter shape" << endl;
11         cin >> name;
12     }
13     virtual void area()
14     {
15         cout << name << endl;
16     }
17     virtual void perimeter()
18     {
19         cout << name << endl;
20     }
21 };
22
23 class rectangle: public shape
24 {
25 public:
26     int length;
27     int breadth;
28     void get()
29     {
30         cout << "length" << endl;
31         cin >> length;
32         cout << "breadth" << endl;
33         cin >> breadth;
34     }
35     void area()
36     {
37         cout << "area of rectangle is" << length * breadth << endl;
38     }
39     void perimeter()
40     {
41         cout << "perimeter of rectangle is" << 2 * (length + breadth) << endl;
42     }
43 };
44
45 class circle: public shape
46 {
47 public:
48     float radius;
49     void get()
50     {
51         cout << "radius" << endl;
52         cin >> radius;
53     }
54     void area()
55     {
56         cout << "area of circle is" << 3.14 * radius * radius << endl;
57     }
58     void perimeter()
59     {
60         cout << "perimeter of circle is" << 2 * 3.14 * radius << endl;
61     }
62 };
63
64 int main()
65 {
66     shape s;
67     s.get();
68     s.area();
69     s.perimeter();
70
71     rectangle r;
72     r.get();
73     r.area();
74     r.perimeter();
75
76     circle c;
77     c.get();
78     c.area();
79     c.perimeter();
80
81     return 0;
82 }
```



```
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 . . .
```