Sir Syed C@SE
Institute of Technology

OOP Lab #9 Polymorphism

SUBMITTED BY:

UBAID AHMAD

ROLL NO:

2410-0011

SUBMITTED TO:

Ma’am Laiba Tanveer

DATE:

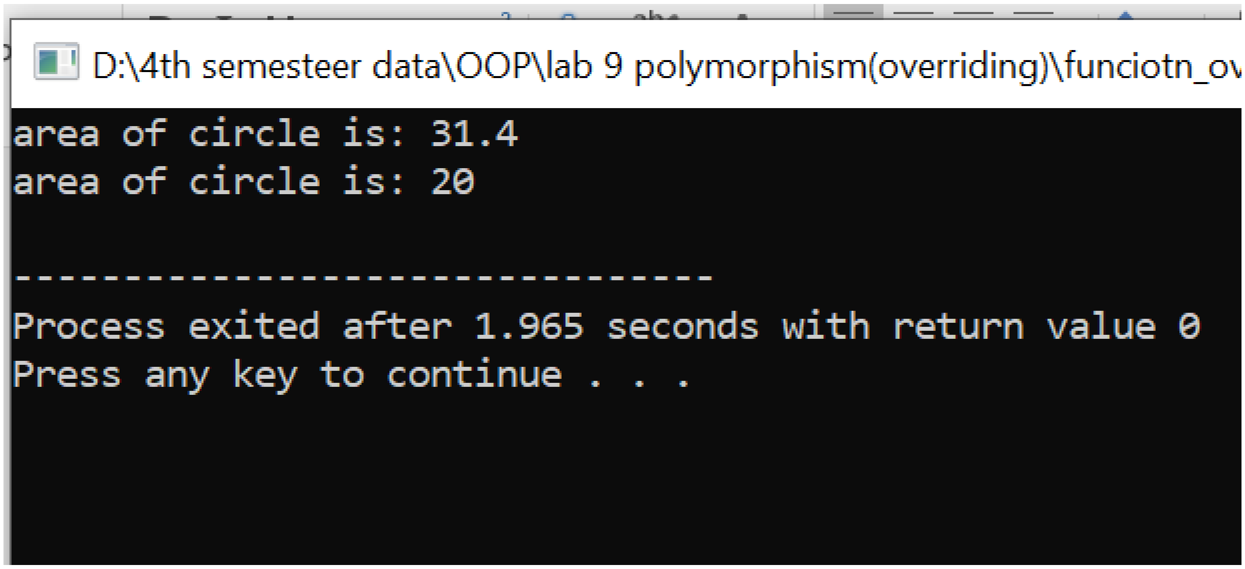
02/12/2025

Task 1

Code:

```
#include <iostream>
using namespace std;
class shape{
    public:
        virtual void calculateArea()=0;
        virtual void displayArea()=0;
};
class circle:public shape{
    int radius;
    double area;
    public:
        circle(int r){
            radius=r;
        }
        void calculateArea(){
            area=2*radius*(3.14);
        }
        void displayArea(){
            cout<<"area of circle is: "<<area<<endl;
        }
};
class rectangle:public shape{
    int length;
    int width;
    double area;
    public:
        rectangle(int l,int w){
            length=l;
            width=w;
        }
        void calculateArea(){
            area=length*width;
        }
        void displayArea(){
            cout<<"area of circle is: "<<area<<endl;
        }
};
int main(){
    shape* pointer;

    circle c(5);  //radius=5
    rectangle r(4,5); //length=4,width=5

    pointer=&c;
    pointer->calculateArea();
    pointer->displayArea();

    pointer=&r;
    pointer->calculateArea();
```

```
        pointer->displayArea();

        return 0;
}
```



D:\4th semesteer data\OOP\lab 9 polymorphism(overriding)\funciotn_ov

```
area of circle is: 31.4
area of circle is: 20

--------------------------------
Process exited after 1.965 seconds with return value 0
Press any key to continue . . .
```

Task 2

Task:

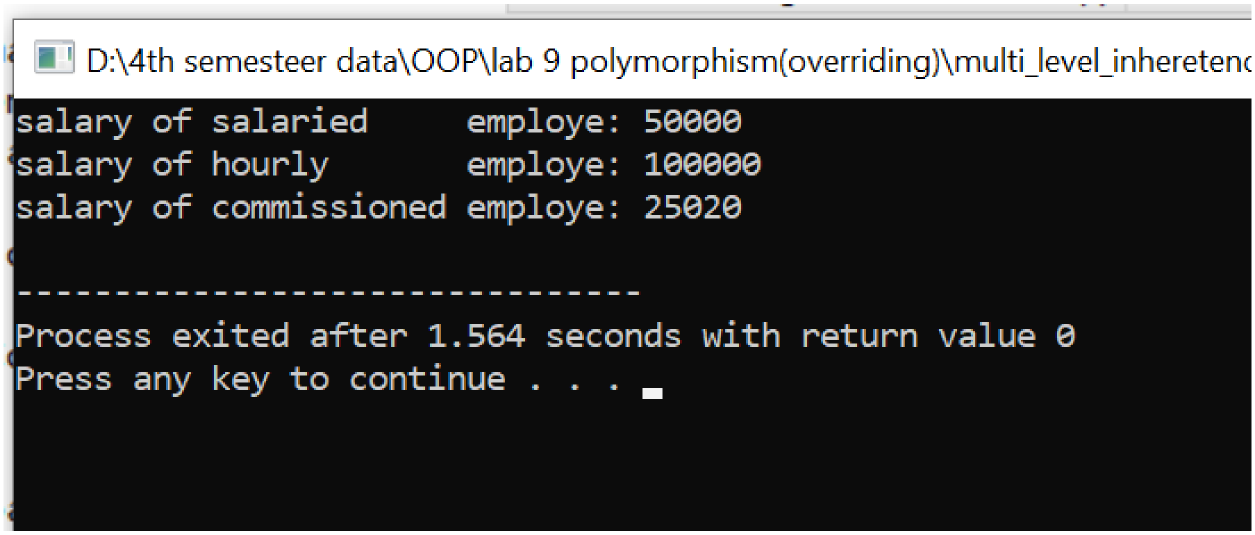
Create Base class employee with constructor virtual float calculatepay().create derived Class Salaried Employee (fixed salary) Hourly Employee(rate per hour and hours worked) and Comission Employee(base salary.+percentage of sales). Use RUNTIME polymorphism by storing different employee objects in base class using pointer. Demonstrate calling calculatepay()through the base class pointer.

CODE:

```
#include <iostream>

using namespace std;

class employee{

        public:

                virtual float calculatePay()=0;
```

```
};
class salariedEmploye:public employee{
    float salary;
    public:
        salariedEmploye(int s){
            salary=s;
        }
        float calculatePay(){
            return salary;
        }
};
class hourlyEmploye:public employee{
    int rate_per_hour;
    int hours_worked;
    public:
        hourlyEmploye(int r,int h){
            rate_per_hour=r;
            hours_worked=h;
        }
        float calculatePay(){
            float salary=rate_per_hour*hours_worked;
            return salary;
        }
};
class CommissionEmploye:public employee{
    int basicSalary;

    int percentageOfsales;

    public:

        CommissionEmploye(int b,int percent){

            basicSalary=b;

            percentageOfsales=percent;

        }

        float calculatePay(){

            float total=basicSalary+percentageOfsales;

            return total;

        }


};

int main(){

    employee* base;

    salariedEmploye salaried(50000);

    hourlyEmploye hourly(5000,20); //hourly rate=5000   hours worked =20


    CommissionEmploye commissioned(25000,20);   //basic salary=25000
percentage of sales=20


base=&salaried;

    float salary1=base->calculatePay();


    base=&hourly;

    float salary2=base->calculatePay();
```

```
        base=&commissioned;
        float salary3=base->calculatePay();

        cout<<"salary of salaried     employe: "<<salary1<<endl;
        cout<<"salary of hourly       employe: "<<salary2<<endl;
        cout<<"salary of commissioned employe: "<<salary3<<endl;

        return 0;
}
```


D:\4th semesteer data\OOP\lab 9 polymorphism(overriding)\multi_level_inheretenc

```
salary of salaried     employe: 50000
salary of hourly       employe: 100000
salary of commissioned employe: 25020

--------------------------------
Process exited after 1.564 seconds with return value 0
Press any key to continue . . .
```

END: