



LABTASK WEEK-8

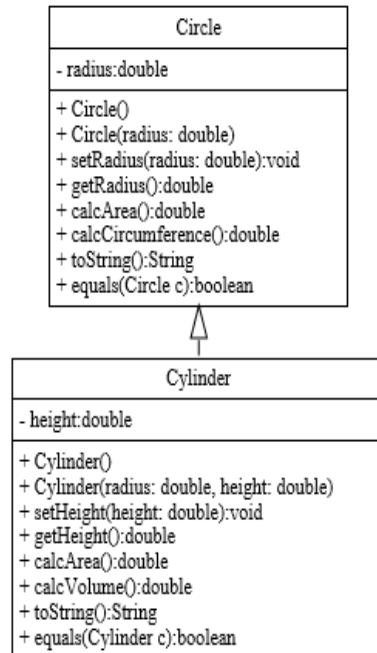
ALI ZIA KHAN

FESE-19052

Inheritance and Method Overriding

Exercise 1

Write the classes below containing the given instance variables and methods, following the inherited hierarchy:



Formulas:

The following formulas may be helpful:

- Area of a circle: πr^2
- Circumference of a circle: $2\pi r$
- Surface area of a cylinder: $2\pi rh + 2\pi r^2$
- Volume of a cylinder: $\pi r^2 h$

Use the following application class to test your code

```
public class Application {
    public static void main(String []args){
        Circle circle1 = new Circle(3);
        Circle circle2 = new Circle(5);
```

```
#include<conio.h>
#include<string.h>
#include<iostream>
#include<cmath>
using namespace std;
class Circle{
public:
    double radius;
public:
    Circle(){

    };
    Circle(double r){
        radius=r;

    }
    void setradius(double r){
        radius=r;
    }
    double getradius(){
        return radius;
    }
    double calcArea(){
        return M_PI * radius*radius;
    }
    double calcCircumference(){

        return 2* M_PI * radius;
    }
    string ToString(){
        |
    }
}
```

```

        bool equals(Circle c){
            }
    };
class Cylinder:public Circle{
public:
    double height;
public:
    Cylinder(){

    };
    Cylinder(double r,double h):Circle(radius){
        height=h;
    };
    void setheight(double h){
        height=h;
    }
    double getheight(){
        return height;
    }
    virtual double calcArea(){
        return (2* M_PI * radius * height) + (2* M_PI* radius * radius );
    }
    double calcVolume(){
        return M_PI*radius*height;
    }
    virtual string ToString(){

    }
    virtual bool equals(Cylinder c){

    }
}

```

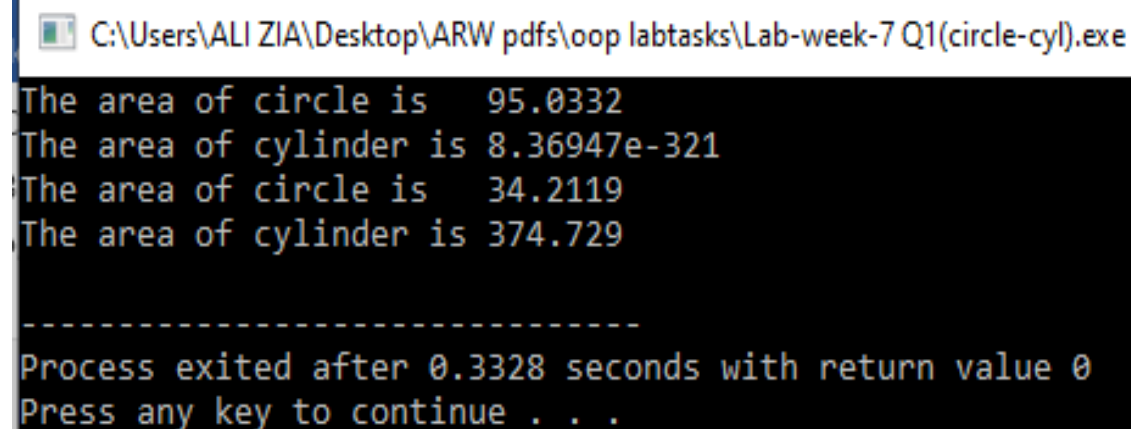
```

};
int main(){
    Circle c1(5.5);
    cout<<"The area of circle is\t"<<c1.calcArea()<<endl;
    Cylinder c2(2.2,3.5);
    cout<<"The area of cylinder is\t"<<c2.calcArea()<<endl;

    Circle *ptr;
    ptr=new Circle;
    ptr->setradius(3.3);
    cout<<"The area of circle is\t"<<ptr->calcArea()<<endl;
    Cylinder c3 ;
    Cylinder *ptr1;
    ptr1=&c3;
    ptr1->setradius(4.2);
    ptr1->setheight(10.0);
    cout<<"The area of cylinder is\t"<<ptr1->calcArea()<<endl;
}

```

OUTPUT:



```

C:\Users\ALI ZIA\Desktop\ARW pdfs\oop labtasks\Lab-week-7 Q1(circle-cyl).exe
The area of circle is 95.0332
The area of cylinder is 8.36947e-321
The area of circle is 34.2119
The area of cylinder is 374.729

-----
Process exited after 0.3328 seconds with return value 0
Press any key to continue . . .

```

Question #2. Apply the Concept of Composition by Creating part classes (doors,windows,Engine,Wheels)and than Crete a whole class Car that is comp osed of all these defined parts.

```
#include<conio.h>
#include<string.h>
#include<iostream>
#include<cmath>
using namespace std;
class Doors {
public:
    double height;
    double width;
    int totaldoors;
public:
    Doors() {
    };
    Doors(double h, double w, int t) :height(h), width(w), totaldoors(t) {
    };
    void showdoors() {
        cout << "the height of doors is" << height << endl;
        cout << "the width of doors is" << width << endl;
        cout << "the no of doors are" << totaldoors << endl;
    }
};
class Windows {
public:
    double totalwindows;
    string color;
    string material;
public:
    Windows() {
    };
};
```

```
Windows(double tw, string c, string m) :totalwindows(tw), color(c), material(m) {
};
void showwindows() {
    cout << "the total no of windows" << totalwindows << endl;
    cout << "the color of windows is" << color << endl;
    cout << "the material is" << material << endl;
}
};
class Engine {
public:
    double horsepower;
    int no_of_pistons;
    string enginetype;
public:
    Engine() {
    };
    Engine(double hp, int p, string et) :horsepower(hp), no_of_pistons(p), enginetype(et) {
    };
    void showengine() {
        cout << "the engine horsepowe" << horsepower << endl;
        cout << "no of pistons" << no_of_pistons << endl;
        cout << "the type of engine is" << enginetype << endl;
    }
};
class Wheels {
public:
    string rim_colors;
    double wheel_radius;
    Wheels(string r, double wr) :rim_colors(r), wheel_radius(wr) {
    };
};
```

```
public:
    Wheels() {
    };
    void showwheels() {
        cout << "the color of rims are" << rim_colors << endl;
        cout << "the radius of wheel are" << wheel_radius << endl;
    }
};

class Car {
public:
    string brand;
    int year;
    string name;
    Doors car_doors;
    Windows car_windows;
    Engine car_engine;
    Wheels car_wheels;
public:
    Car() {
    };
    Car(string b, int y, string n, Doors d1, Windows w1, Engine e1, Wheels w1) {
        brand = b;
        year = y;
        name = n;
        car_doors = d1;
```



```

        car_windows = w1;
        car_engine = e1;
        car_wheels = w1;


    };

    void showdata() {
        cout << "the car brand is" << brand << endl;
        cout << "the car year is" << year << endl;
        cout << "The car name is" << name << endl;
        cout << "the door details are" << endl;
        car_doors.showdoors();
        cout << "the window details are" << endl;
        car_windows.showwindows();
        cout << "the engine details are" << endl;
        car_engine.showengine();
        cout << "the wheels details are" << endl;
        car_wheels.showwheels();
    }
};

int main() {
    Doors d1(6.1, 5.5, 4);
    Windows w1(4, "blue", "glass");
    Engine e1(55.5, 2, "Thermal");
    Wheels w1("pink", 3.2);
    Car c1("toyota", 2019, "corolla", d1, w1, e1, w1);
    c1.showdata();
}

```

OUTPUT:

 Microsoft Visual Studio Debug Console

```
the car brand istoyota
the car year is2019
The car name iscorolla
the door details are
the height of doors is6.1
the width of doors is5.5
the no of doors are4
the window details are
the total no of windows4
the color of windows isblue
the material isglass
the engine details are
the engine horsepowe55.5
no of pistons2
the type of engine isThermal
the wheels details are
the color of rims arepink
the radius of wheel are3.2
```

Question #3.

Create Box class containing data members (width ,height,Length) and Member functions showdata and Volume.. Apple the concept of Operator Overloading and overload operators (+,-,* ,/) for addition subtraction multiplication and division of Objects of Box class.

```
#include<string.h>
#include<conio.h>
#include<iostream>
using namespace std;
class Box {
public:
    double height;
    double width;
    double length;
public:
    Box() {
    };
    Box(int h, int w, int l) :height(h), width(w), length(l) {
    };
    void setheight(double h) {
        height = h;
    }
    void setwidth(double w) {
        width = w;
    }
    void setlength(double l) {
        length = l;
    }
    double getlength() {
        return length;
    }
    double getwidth() {
        return width;
    }
}
```

```

double getheight() {
    return height;
}
double volume() {
    return height * length * width;
}
Box operator+(const Box& b) {
    Box box;
    box.height = this->height + b.height;
    box.width = this->width + b.width;
    box.length = this->length + b.length;

    return box;
}
Box operator/(const Box& b) {
    Box box;
    box.height = this->height / b.height;
    box.width = this->width / b.width;
    box.length = this->length / b.length;

    return box;
}
Box operator-(const Box& b) {
    Box box;
    box.height = this->height - b.height;
    box.width = this->width - b.width;
    box.length = this->length - b.length;

    return box;
}

```

```

Box operator*(const Box& b) {
    Box box;
    box.height = this->height * b.height;
    box.width = this->width * b.width;
    box.length = this->length * b.length;

    return box;
}

void showdata(Box b3) {
    cout << "Details of box b3 are\n\n" << endl;
    cout << "Height of b3 is\t" << height << "m" << endl;
    cout << "Width of b3 is\t" << width << "m" << endl;
    cout << "Length of b3 is\t" << length << "m" << endl;
}

};
int main() {

    Box b1(5.6, 6.4, 7.2);
    Box b2(3.5, 4.8, 2.2);
    Box b3;

    b3 = b1 + b2;

    b3.showdata(b3);
    cout << "This result was by using + operator" << endl;
    cout << "The volume is \t" << b3.volume() << "m^3\n\n" << endl;

    b3 = b1 * b2;

    b3.showdata(b3);
    cout << "This result was by using * operator" << endl;
    cout << "The volume is \t" << b3.volume() << "m^3\n\n" << endl;

```

```

    b3 = b1 - b2;
    cout << "using - operator" << endl;

    b3.showdata(b3);
    cout << "This result was by using - operator" << endl;
    cout << "The volume is \t" << b3.volume() << "m^3\n\n" << endl;

    b3 = b1 / b2;

    b3.showdata(b3);
    cout << "This result was by using / operator" << endl;
    cout << "The volume is \t" << b3.volume() << "m^3\n\n" << endl;

    _getch();
    return 0;
}

```

OUTPUT:

```
Height of b3 is 8m
Width of b3 is 10m
Length of b3 is 9m
This result was by using + operator
The volume is 720m^3
```

Details of box b3 are

```
Height of b3 is 15m
Width of b3 is 24m
Length of b3 is 14m
This result was by using * operator
The volume is 5040m^3
```

using - operator
Details of box b3 are

```
Height of b3 is 2m
Width of b3 is 2m
Length of b3 is 5m
This result was by using - operator
The volume is 20m^3
```

Details of box b3 are

```
Height of b3 is 1.66667m
Width of b3 is 1.5m
Length of b3 is 3.5m
This result was by using / operator
The volume is 8.75m^3
```