|  |
| --- |
| Armghan Ahmad |
| OOP LAB 13 LAB WORK |
| 20P-0183 |

|  |
| --- |
| Section :2B  6/11/2021 |

Question 3:

#include <iostream>

using namespace std;

class Shape {

    private:*// private access speccifier*

        float size=20;

    protected:*// protected  access specifier*

        float area=34;

    public:*// public  access modifier*

        float volume = 33;

        void draw() {

            cout<< "Parent Class: Shape:";

        }

        void dispalySize() {

            cout<< "Parent Class: size:"<<size;

        }

};

class Cricle : private Shape {*// public inheri...*

    public:

    void draw() {

        cout<< "Child Class: Circle:"<<endl;

*// because its public Inheritance we can access  (public, procted) member*

*// with out creating object in child class*

        cout<< "Partent Class Area:" << area<<endl;*// proctected*

        cout<< "Partent Class Volume:" << volume<<endl;*// public*

        dispalySize();

    }

};

int main () {

    Cricle circle;

*// We are accessing public variable of Parent class though child class object*

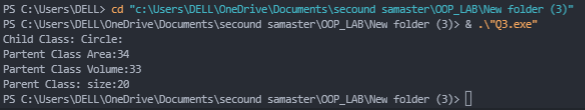
*// cout<< "Shape Volume"<< circle.volume<<endl;*

    circle.draw();

    return 0;

}

Output:



Question 2:

#include <iostream>

using namespace std;

class Shape {

    private:*// private access speccifier*

        float size=20;

    protected:*// protected  access specifier*

        float area=34;

    public:*// public  access modifier*

        float volume = 33;

        void draw() {

            cout<< "Parent Class: Shape:";

        }

        void dispalySize() {

            cout<< "Parent Class: size:"<<size;

        }

};

class Cricle : protected Shape {*// public inheri...*

    public:

    void draw() {

    cout<< "Child Class: Circle:"<<endl;

*// because its public Inheritance we can access  (public, procted) member*

*// with out creating object in child class*

    cout<< "Partent Class Area:" << area<<endl;*// proctected*

    cout<< "Partent Class Volume:" << volume<<endl;*// public*

    dispalySize();

    }

};

int main () {

    Cricle circle;

*// We are accessing public variable of Parent class though child class object*

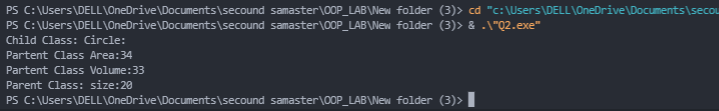
*// cout<< "Shape Volume"<< circle.volume<<endl;*

    circle.draw();

    return 0;

}

Output:



Question 1:

#include <iostream>

using namespace std;

class Shape {

    private:*// private access specifire*

        float size=20;

    protected:*// protected  access specifire*

        float area=34;

    public:*// public  access modifire*

        float volume = 33;

        void draw() {

            cout<< "Parent Class: Shape:";

        }

        void dispalySize() {

            cout<< "Parent Class: size:"<<size;

        }

};

class Cricle : public Shape {*// public inheri...*

    public:

    void draw() {

    cout<< "Child Class: Circle:"<<endl;

*// because its public Inheritance we can access  (public, procted) member*

*//with out creating object in child class*

    cout<< "Partent Class Area:" << area<<endl;*// proctected*

    cout<< "Partent Class Volume:" << volume<<endl;*// public*

    dispalySize();

    }

};

int main () {

    Cricle circle;

*// We are accessing public variable of Parent class though child class object*

    cout<< "Shape Volume"<< circle.volume<<endl;

    circle.draw();

    return 0;

}

Output:

