Short Questions

1. **class A: public class B { }**

State which members (public/protected/private) of class A can be accessed from class B.

A is a derived class. Bass class cannot access any member of the Derived class

2. **Class Square is derived from class Rectangle. Write the constructor (one argument) of class Square and call the (two arguments) constructor of Rectangle.**

```
class Square: public Rectangle{
private:
   int length;
public:
  Square(int x, int y, int l):Rectangle(x, y), length(l){ }
};
```

3. **Are overloaded operators inherited in the derived class?**

Yes, overloaded operators can be inherited in the derived class.

4. **Consider the following code:**

```
class A {
public:
A ( )
{cout<<"A:default"<<endl;}
A (int a)
{cout<<"A:parameter"<<endl;}
};
class B : public A
{
public:
B (int a)
{cout<<"B"<<endl;}
};
```

Write the output of:

B b(2);

B b2;

A: default

B

A: default

5. **The class B in the above question is changed as follows:**

class B : public A

{

public:

B (int a): A(a)

{cout<<"B"<<endl;}

};

What is the output of:

B test(5);

A: parameter

B

Exercise-01

Create a class Point with two data members x and y. Provide appropriate constructors, get, set and display methods. Derive a class Circle from Point. The circle, in addition to the center (Point) also has radius as its data member. Provide constructors, get and set methods in the circle class. Also provide methods to compute area and circumference of the circle. (Area of circle is PI * r * r and Circumference is: 2 * PI * r). Derive a class Cylinder from circle with a data member height. Provide constructors and set/get methods to set/get height of the cylinder. Provide a function to compute area of cylinder. (A = 2 x area of circle + 2*PI*r*height). Also provide a function to compute the volume of cylinder (PI * r*r*h). Create an object of class Cylinder and compute its volume and surface area. Print your name, reg,no, section, semester in main using cout statements.

Code:

```
#include "stdafx.h"
#include<iostream>
using namespace std;
```

```
const float pi=3.14;
class Point{
protected:
	int x,y;

public:

	Point():x(0),y(0){}

	Point(int n,int m):x(n),y(m){}

	void setx(int n){
	 x=n;  }

	void sety(int m){
		y=m;}

	int getx(){
		return x;}

	int gety(){
		return y;}

	void display(){
		cout<<"x= "<<x<<"y= "<<y<<endl;}

};

class Circle:public Point{
protected:
	float r;  //radius
public:
	Circle():Point(),r(0){};
	Circle(int n,int m,float o):Point(n,m),r(o){};

	void setr(int o){
	 r=o;  }

	float getr(){
		return r;}

	float Cirarea(){
		return pi*r*r;
	}

	void Circumference(){
		cout<<"Circumference of Circle = "<<2*pi*r<<endl;
	}

};

class Cylinder:public Circle{
private:
	float h;  //height
```

```
public:
	Cylinder():Circle(),h(0){};
	Cylinder(int n,int m,float o,int p):Circle(n,m,o),h(p){};

	void seth(int p){
	 h=p;  }

	float geth(){
		return h;}

	void Cylarea(){
		cout<<"Area of Cylinder = "<<2*Cirarea()+2*pi*h<<endl;
	}

	void Volume(){
		cout<<"Volume of cylinder = "<<pi*r*r*h<<endl;
	}
};

int _tmain(int argc, _TCHAR* argv[])
{
	Cylinder cy(3,5,6.2,7.8);
	cy.Cylarea();
	cy.Volume();
	cout<<"Sobia Karim\n2022-BSE-069\nsemseter:2 B"<<endl;
	system("pause");
	return 0;
}
```

Output:

```
Area of Cylinder = 285.363
Volume of cylinder = 844.911
Sobia Karim
2022-BSE-069
semseter:2 B
Press any key to continue . . .
```

Exercise-02

Write the above program by creating a separate (header+source) file for each of the classes. Therefore, you will have 3 header and 3 source files for the three classes and one source file for your main program. Refer to the lab manual to see details on how you add classes to your project.

Code:

Main:

```
#include "stdafx.h"
#include "Pointt.h"
#include "Circle.h"
#include "Cylinder.h"
#include<iostream>
using namespace std;
const float pi=3.14;


int _tmain(int argc, _TCHAR* argv[])
{
	Cylinder cy(3,5,6.2,7.8);
	cy.Cylarea();
	cy.Volume();
	cout<<"Sobia Karim\n2022-BSE-069\nsemseter:2 B"<<endl;
	system("pause");
	return 0;
}
```

Point.h

```
#pragma once
class Pointt
{
protected:
	int x,y;

public:

	Pointt();

	Pointt(int n,int m);

	void setx(int n);

	void sety(int m);

	int getx();

	int gety();

	void display();
};
```

Point.cpp

```
#include "StdAfx.h"
#include "Pointt.h"
#include <iostream>
using namespace std;

	Pointt::Pointt():x(0),y(0){}

	Pointt::Pointt(int n,int m):x(n),y(m){}

	void Pointt::setx(int n){
	 x=n;  }

	void Pointt::sety(int m){
		y=m;}

	int Pointt::getx(){
		return x;}

	int Pointt::gety(){
		return y;}

	void Pointt::display(){
		cout<<"x= "<<x<<"y= "<<y<<endl;}
```

Circle.h

```
#pragma once
#include "pointt.h"
class Circle :
	public Pointt
{
protected:
	float r;  //radius

public:
	Circle();

	Circle(int n,int m,float o);

	void setr(int o);

	float getr();

	float Cirarea();

	void Circumference();
};
```

Circle.cpp

```
#include "StdAfx.h"
```

```
#include "Circle.h"
#include <iostream>
using namespace std;
const float pi=3.14;


	Circle::Circle():Pointt(),r(0){};
	Circle::Circle(int n,int m,float o):Pointt(n,m),r(o){};

	void Circle::setr(int o){
	 r=o;  }

	float Circle::getr(){
		return r;}

	float Circle::Cirarea(){
		return pi*r*r;
	}

	void Circle::Circumference(){
		cout<<"Circumference of Circle = "<<2*pi*r<<endl;
	}
```

Cylinder.h

```
#pragma once
#include "circle.h"
class Cylinder :
	public Circle
{
private:
	float h;  //height
public:
	Cylinder();
	Cylinder(int n,int m,float o,int p);

	void seth(int p);

	float geth();

	void Cylarea();

	void Volume();
};
```

Cylinder.cpp

```
#include "StdAfx.h"
#include "Cylinder.h"
#include <iostream>
using namespace std;
const float pi=3.14;

	Cylinder::Cylinder():Circle(),h(0){};
	Cylinder::Cylinder(int n,int m,float o,int p):Circle(n,m,o),h(p){};
```

```
void Cylinder::seth(int p){
 h=p;  }

float Cylinder::geth(){
       return h;}

void Cylinder::Cylarea(){
       cout<<"Area of Cylinder = "<<2*Cirarea()+2*pi*h<<endl;
}

void Cylinder::Volume(){
       cout<<"Volume of cylinder = "<<pi*r*r*h<<endl;
}
```

Output: