```
// NAME : LEWIS MUNENE KIRIMI
// REG NO . : SCT211-0014/2021
// UNIT CODE : ICS2201
// ASSIGNMENT 1
#include<iostream>
#include<string>


using namespace std;
const double PI = 3.14159265358979323846;
class Square{
private:
double side;
public:
// constructors in the class
// default constructor
Square(){
side = 0;
}
// parameterized constructor
Square(double s)
{
side = s;
}
// setter
void setSide(int s)
{
side = s;
}
// getter
double getSide()
{
return side;
}
// compute area
double computeArea()
{
return side * side;
}
// compute perimeter
double computePerimeter()
{
return 4 * side;
}
// display
void printSquare()
{
cout << "\n\n\n Shape : Square \n\nLength :\t"<< getSide() <<
"cm\nArea :\t"<<computeArea() << " cm squared\nPerimeter :\t"<<
computePerimeter()<<"cm"<<endl;
}
};
class Circle{
private:
double radius;
public:
// default constructor
Circle(){
radius = 0;
```

```
}
// parameterized constructor
Circle(double r)
{
radius = r;
}
// setter
void setRadius(double r)
{
radius = r;
}
// getter
double getRadius()
{
return radius;
}
// compute area
double getArea()
{
return PI * radius * radius;
}
// compute perimeter
double computePerimeter()
{
return 2 * PI * radius;
}
// display
void printCircle()
{
cout << "\n\n\n Shape : Circle \n\nRadius :\t"<< getRadius() << "cm\nArea :\t"<<getArea()
<< " cm squared\nPerimeter :\t"<< computePerimeter()<<"cm"<<endl;
}
};
class Cylinder{
private:
double radius;
double height;
public:
Cylinder()
{
radius = 0;
height = 0;
}
Cylinder(double r, double h )
{
radius = r;
height = h;
}
double computeArea()
{
return 2 * PI * radius * (radius + height);
}
double computeVolume()
{
return PI * radius * radius * height;
}
void setRadius(double r)
{
radius = r;
```

```
}
double getRadius()
{
return radius;
}
void setHeight(double h)
{
height = h;
}
double getHeight()
{
return height;
}
void printCylinder()
{
cout << "\n\n\n Shape : Cylinder \n\nRadius :\t"<< getRadius() <<
"cm\nHeight :\t"<< getHeight() << "cm\nArea :\t"<< computeArea() << "
cmsquared\nVolume :\t"<< computeVolume()<<"cm cubed"<<endl;
}
};
class Sphere{
private:
double radius;
public:
Sphere(){
radius = 0;
}
Sphere(double r)
{
radius = r;
}
double computeArea()
{
return 4 * PI * radius * radius;
}
double computeVolume()
{
return 4/3 * PI * radius * radius * radius;
}
void setRadius(double r)
{
radius = r;
}
double getRadius()
{
return radius;
}
void printSphere()
{
cout << "\n\n\n Shape : Sphere \n\nRadius :\t"<< getRadius() <<
"cm\nArea :\t"<<computeArea() << " cm squared\nVolume :\t"<<
computeVolume()<<"cmcubed"<<endl;
}
};
int main()
{
// displaying the name and registration number of a student
cout << "Name:\tLEWIS MUNENE KIRIMI \n";
cout << "Reg. No.:\tSCT211-0014/2021";
cout << "\nAssignment2\nUnit code \t ICS2201 \nUnit Name :";
cout << " OBJECT ORIENTED PROGRAMMING 2"<<endl;
// declaring objects of the classes Square, Circle, Cylinder and Sphere
Square square;
Circle circle;
Cylinder cylinder;
Sphere sphere;
int choice;
cout << "\n\n\n Choose a Figure below:";
cout << " \n1.Square \n2.Circle";
cout <<"\n3.Cylinder\n4.Sphere\n5.Exit \n\n >>>>" ;
// getting the input choice
cin >> choice;
switch (choice)
{
case 1:
cout << "Enter the length of the square in centimetres : ";
double side;
cin >> side;
square.setSide(side);
square.printSquare();
break;
case 2:
cout << "Enter the radius of the circle in centimetres : ";
double circleRadius;
cin >> circleRadius;
circle.setRadius(circleRadius);
circle.printCircle();
break;
case 3:
double cylinderRadius;
double cylinderHeight;
cout << "Enter the radius of the cylinder in centimetres : ";
cin >> cylinderRadius;
cout << "Enter the height of the cylinder in centimetres : ";
cin >> cylinderHeight;
cylinder.setRadius(cylinderRadius);
cylinder.setHeight(cylinderHeight);
cylinder.printCylinder();
break;
case 4:
double sphereRadius;
cout << "Enter the radius of the sphere in centimetres : ";
cin >> sphereRadius;
sphere.setRadius(sphereRadius);
sphere.printSphere();
break;
case 5:
cout << "Bye Bye ! See you next time !";
exit(0);
break;
default:
cout<<"Invalid choice"<<endl;
break;
}
}
```