

## 2B. class Shape, Circle, Cylinder, Polygon, PolygonPillar

### 2B.1 class Shape

```
/* Shape.h */
. . . . // 필요한 헤더파일 내용 추가
class Shape
{
public:
    Shape(); // default constructor
    Shape(string name); // constructor
    virtual void print(ostream &);
protected:
    string name;
};
```

- print(), get\_area(), get\_volume() 함수는 가상함수로 선언되어 late-binding이 실행됨
- class Shape관련 멤버함수들은 Shape.h에 inline 함수로 구현 또는 Shape.cpp 파일에 포함시킬 것

### 2B.2 class Circle

```
/* Circle.h */
. . . . // 필요한 헤더파일 내용 추가
class Circle : public Shape
{
public:
    Circle(string name, double radius);
    double get_circumference(); // 원둘레
    double get_area(); // 원의 면적
    virtual void print(ostream &);
protected:
    int radius;
};
```

- get\_circumference() 멤버함수는 원의 둘레 길이를 계산하여 반환
- get\_area() 멤버함수는 원의 면적을 계산하여 반환
- print() 함수는 원의 기본 정보 (반지름)와 원둘레 및 원면적을 출력
- class Circle관련 멤버함수들은 Circle.h에 inline 함수로 구현 또는 Circle.cpp 파일에 포함시킬 것

### 2B.3 class Cylinder

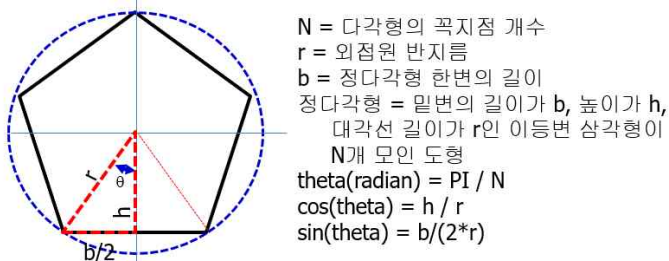
```
/* Cylinder.h */
. . . . // 필요한 헤더파일 내용 추가
class Cylinder : public Circle
{
public:
    Cylinder(string n, double radius, double height);
    double get_circumference(); // perimeter of cross section
    double get_area(); // 원기둥의 표면적
    double get_volume(); // 원기둥의 체적
    virtual void print(ostream &);
protected:
    int height; // Cylinder height
};
```

- get\_circumference() 함수는 원기둥의 둘레 (원의 둘레) 길이를 계산하여 반환
- get\_area() 멤버함수는 원기둥의 표면적 (surface area)을 계산하여 반환
- print() 함수는 원기둥의 기본 정보 (반지름, 높이)와 단면적 (x-area), 원둘레, 표면적 및 체적을 출력
- class Cylinder관련 멤버함수들은 Cylinder.h에 inline 함수로 구현 또는 Cylinder.cpp 파일에 포함시킬 것

## 2B.4 class Polygon

```
/* Polygon.h */
. . . . // 필요한 헤더파일 내용 추가
class Polygon : public Shape
{
public:
    Polygon(string name, double radius, int num_poly);
    double get_base_length();
    double get_perimeter(); // 정다각형 둘레
    double get_area(); // 정다각형 면적
    double get_volume() { return 0.0; }
    void print(ostream &);
protected:
    int radius; // 외접 원의 반지름
    int num_vertices;
};
```

- get\_base\_length() 멤버함수는 반지름 radius에 내접하는 정다각형의 한 변의 길이 (base\_length)를 계산하여 반환할 것.
- get\_perimeter() 멤버함수는 정다각형 둘레 길이를 계산하여 반환
- 정다각형의 둘레길이, 면적 계산에서의 고려사항 (5각형의 예):



- get\_area() 멤버 함수는 다각형 면적을 계산하여 반환할 것. (Hint: 주어진 반지름 r의 원에 내접하는 정다각형의 면적 계산에서는 이등변 삼각형이 꼭지점 개수 만큼 있는 도형을 먼저 생각하고, 이등변 삼각형의 밑변의 길이와 높이를 계산하여 이등변 삼각형의 넓이를 구하고, 이 값에 꼭지점 개수를 곱하여 계산하는 방법을 사용할 수 있음.)
- print() 멤버함수는 다각형의 기본 정보들 (이름, 외접원의 반지름, 꼭지점 개수, 다각형 한 변의 길이)과 면적 (area)을 출력할 것.
- 모든 실수 (double) 자료형의 출력에서는 소숫점 이하 2자리까지 출력할 것.
- class Polygon관련 멤버함수들은 Polygon.h에 inline 함수로 구현 또는 Polygon.cpp 파일에 포함시킬 것

## 2B.5 class PolygonPillar

```
/* PolygonPillar.h */
. . . . // 필요한 헤더파일 내용 추가
class PolygonPillar : public Polygon
{
public:
    PolygonPillar(string n, double radius, int num_vert, double height);
    double get_xperimeter(); // perimeter of cross section
    double get_area(); // 정다각형 기둥의 단면적
    double get_volume(); // 정다각형 기둥의 표면적
    virtual void print(ostream&);
protected:
    int height; // PolygonPillar height
};
```

- get\_xperiment() 멤버함수는 정다각형 기둥의 둘레 (즉, 정다각형의 둘레) 길이를 계산하여 반환
- get\_area() 멤버 함수는 표면적을 계산하여 반환할 것
- get\_volume() 멤버함수는 3차원 도형의 체적(volume)을 계산하여 반환할 것

- print() 멤버함수는 다각형 기둥의 기본 정보들 (이름, 외접원의 반지름, 꼭지점 개수, 다각형 한 변의 길이) 과 단면적 (cross-sectional area), 표면적 (surface area) 및 체적 (volume)을 출력할 것.
- class PolygonPillar 관련 멤버함수들은 PolygonPillar.h에 inline 함수로 구현 또는 PolygonPillar.cpp 파일에 포함시킬 것

## 2B.6 main() 함수

```
/* main.cpp */
...// 필요한 헤더파일 내용 추가 및 전처리기 선언
int main()
{
    Circle cir("Circle", 10);
    Polygon plg_3("Triangle", 10, 3);
    Polygon plg_4("Square", 10, 4);
    Polygon plg_5("Pentagon", 10, 5);
    Polygon plg_6("Hexagon", 10, 6);
    Cylinder cyl("Cylinder", 10, 20);
    PolygonPillar plyPlr_3("Prism", 10, 3, 20);
    PolygonPillar plyPlr_4("Hexahedron", 10, 4, 20);
    PolygonPillar plyPlr_5("PentagonPillar", 10, 5, 20);
    PolygonPillar plyPlr_6("HexagonPillar", 10, 6, 20);

    Shape* shapes[NUM_SHAPES];

    shapes[0] = &cir;
    shapes[1] = &plg_3;
    shapes[2] = &plg_4;
    shapes[3] = &plg_5;
    shapes[4] = &plg_6;
    shapes[5] = &cyl;
    shapes[6] = &plyPlr_3;
    shapes[7] = &plyPlr_4;
    shapes[8] = &plyPlr_5;
    shapes[9] = &plyPlr_6;

    for (int i = 0; i < NUM_SHAPES; i++)
    {
        shapes[i]->print(cout);
        cout << endl;
    }

    return 0;
} // end of main()
```

## 2B.7 실행 결과 예시

```
Circle: radius (10), circumference ( 62.83), area(314.16)
Triangle: radius (10), num_vertices (3), base_length (17.32, perimeter (51.96), area(129.90)
Square: radius (10), num_vertices (4), base_length (14.14, perimeter (56.57), area(200.00)
Pentagon: radius (10), num_vertices (5), base_length (11.76, perimeter (58.78), area(237.76)
Hexagon: radius (10), num_vertices (6), base_length (10.00, perimeter (60.00), area(259.81)
Cylinder: radius (10), circumference ( 62.83), x-area(314.16), height (20), surface area ( 1884.96), volume ( 6283.18)]
Prism: radius (10), base_length (17.32), x-perimeter (51.96), x-area (129.90), height (20), surface area ( 1299.04), volume ( 2598.08)]
Hexahedron: radius (10), base_length (14.14), x-perimeter (56.57), x-area (200.00), height (20), surface area ( 1531.37), volume ( 4000.00)]
PentagonPillar: radius (10), base_length (11.76), x-perimeter (58.78), x-area (237.76), height (20), surface area ( 1651.10), volume ( 4755.28)]
HexagonPillar: radius (10), base_length (10.00), x-perimeter (60.00), x-area (259.81), height (20), surface area ( 1719.61), volume ( 5196.15)]
```

## 2B.8 결과물 제출

- 바탕화면에 Exam2 폴더내에 Exam2B\_학번\_이름 프로젝트를 생성
- 압축 파일 내에 포함사항 : 작성한 프로젝트, 실행결과 Capture(채점 시 정확한 실행 유무를 확인하기 위함)
- 실행 화면 캡처파일은 각 시험 섹션별로 프로젝트 폴더 내에 저장 후 시험 섹션별 폴더별로 압축
- 제출시 .vs 폴더는 삭제 후 문제별 폴더를 압축하여 제출