10802 CPP Midterm Exam

Contributor：Wei-Hsuan Liu

Subject：Train Volume

Main testing concept：

Basics

■ C++ BASICS
□ FLOW OF CONTROL
■ FUNCTION BASICS
□ PARAMETERS AND OVERLOADING
□ ARRAYS
■ STRUCTURES AND CLASSES
□ CONSTRUCTORS AND OTHER TOOLS
□ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES
□ STRINGS
□ POINTERS AND DYNAMIC ARRAYS

Functions

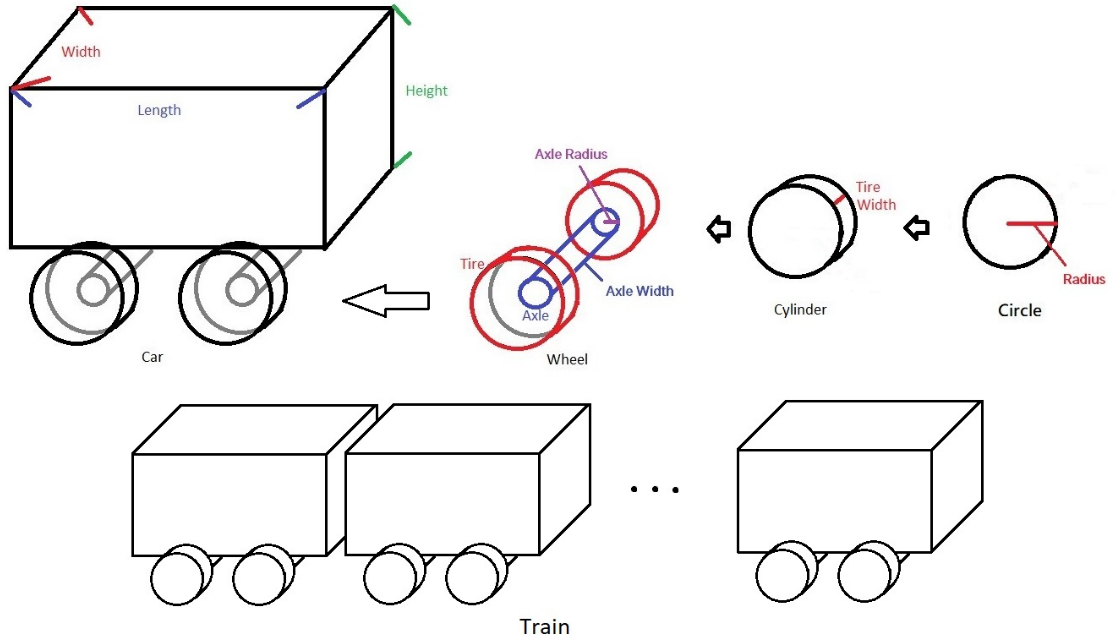
□ SEPARATE COMPILATION AND NAMESPACES
□ STREAMS AND FILE I/O
□ RECURSION
□ INHERITANCE
□ POLYMORPHISM AND VIRTUAL FUNCTIONS
□ TEMPLATES
□ LINKED DATA STRUCTURES
□ EXCEPTION HANDLING
□ STANDARD TEMPLATE LIBRARY
□ PATTERNS AND UML

Description：

In this task, we simply assume that a train is composed of 2 parts, cars and wheels. This task asks you to calculate the volume of a train by defining 3 component classes and 2 shape classes: **Train**, **Car, Wheel, Cylinder and Circle**.

A **Train** has multiple **cars**. A **Car** has a **cuboid**(長方體) and multiple **wheels**. A **Wheel** has two **tires** and one **axle**. The **tire** and the **axle** are both **Cylinders**. A **Cylinder** is made up of a **Circle**. Note that the volume of a **cuboid** is **width * height * length**, and the area of a **circle** is **radius * radius * 3.14159**. None of the parts from the train are overlapped.

The following figures shows the composition of basic shapes including circle and to the detail train structure is shown below.



To be specific, all classes need to implement following functions:

```
        void input();                //Get input
        double getVolume();    //Return class volume
```

You have to run the main function down below correctly and **can’t change any code of it**.

```
int main(){
        Train train;
        train.input();
        std::cout << train.getVolume() << std::endl;
        return 0;
}
```

Input：

The input contains several sets of train attributes. Each set has the input lines as described below.

1. The first line contains an *integer* **<car_count>** indicating the number of cars.
2. The second line contains 3 *doubles* **<car_length> <car_width> <car_height>** indicating the car’s length, width, and height respectively.
3. The third line contains an *integer* **<wheel_count>** indicating the number of wheels.
4. The fourth line contains 2 *doubles* **<tire_radius> <tire_width>** indicating the tire’s radius and width.
5. The fifth line contains 2 *doubles* **<axle_radius> <axle_width>** indicating the axle’s radius and width.

There will be no exception to Input.

Output：

The volume of the Train (rounded down to the nearest *integer*).

Sample Input / Output :

Sample Input	**Sample Output**

6
100 30 40
5
5 1
1 29
	727445

13
71 31 29
9
4 2
2 31
	898879

□ **Easy, only basic programming syntax and structure are required.**

■ **Medium, multiple programming grammars and structures are required.**

□ **Hard, need to use multiple program structures or complex data types.**

Expected solving time:

20 minutes

Other notes：

The length unit is meter.
The volume unit is square meter.