OBJECT ORIENTED PROGRAMMING 1 LABORATORY
Experiment # 2:
Introduction to Classes and Objects

OBJECTIVES

The main purpose of this experiment is to introduce you to classes and objects. In this experiment, firstly, steps for creating the class interface, class implementation, and driver program are given. Then, a simple example is studied. Lastly, students will modify the example to enhance the given class.

INFORMATION

IDEs are used to develop, compile and, run a computer program. Dev-C++ is a well-known, free, fast, and simple IDE for developing C++ codes. In order to use the IDE, you must follow the steps that are given below.

Step 1: Create a folder named studentNumber_NameSurname_xA in D and save all files in this folder.

Step 2: Run the Dev-C++. Open a new project by using the icon as shown in Figure 1.

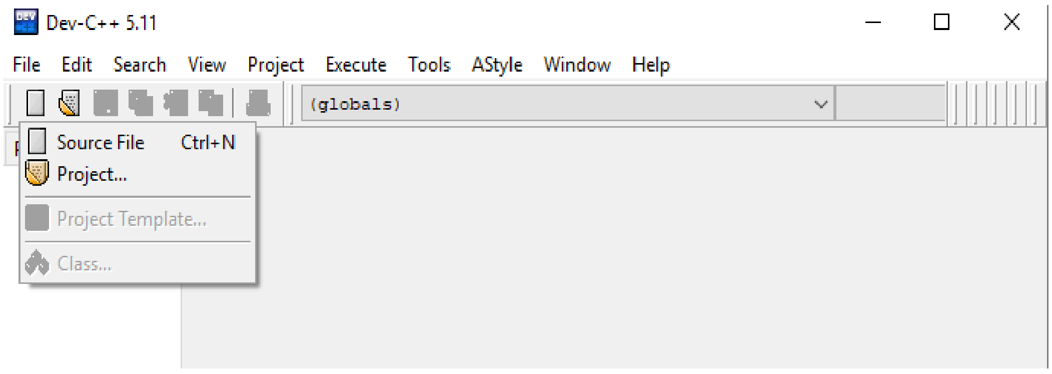
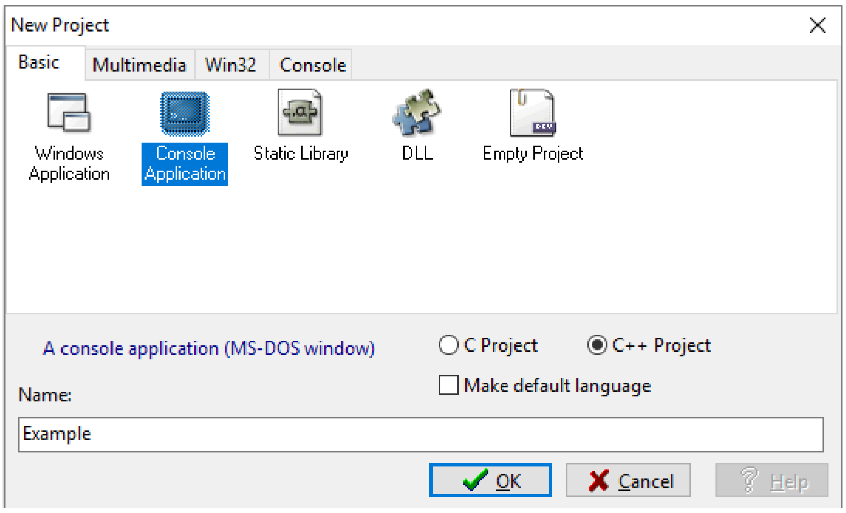


Fig. 1 Open a new project

Step 2: Then, select Console Application and enter the project name as shown in Figure 2.



A console application (MS-DOS window)

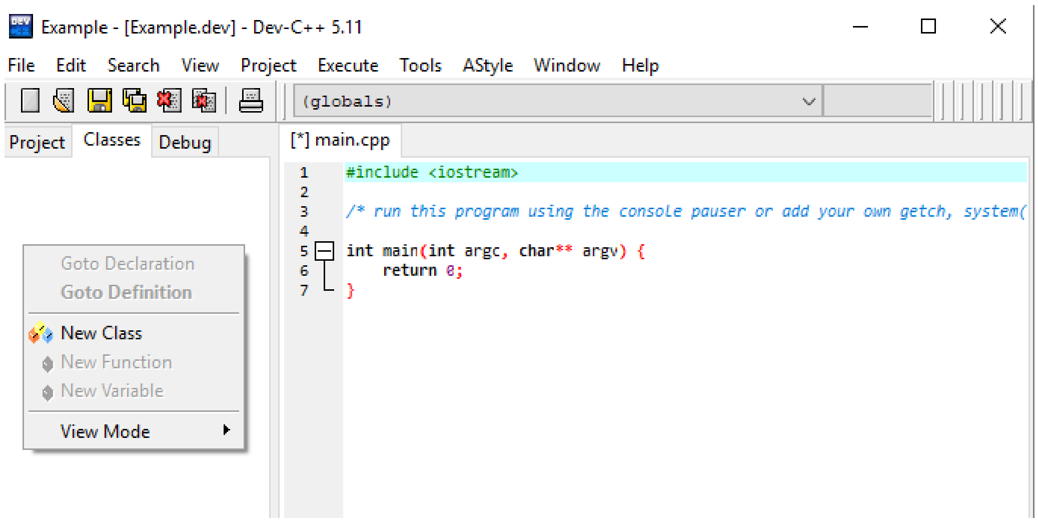
C Project [✓ C++ Project]

☐ Make default language

Name: Example

Fig. 2 Save the project

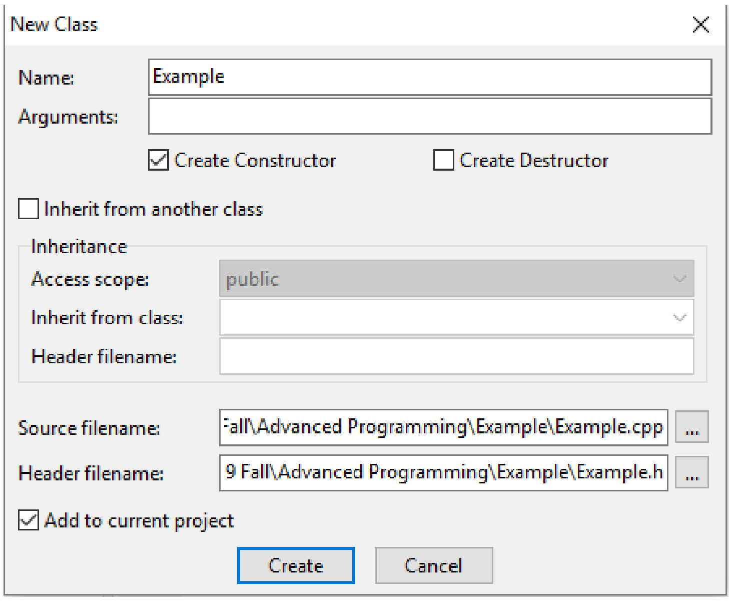
Step 3: Create a new class as shown in Figure 3.



```
#include <iostream>

/* run this program using the console pauser or add your own getch, system(

int main(int argc, char** argv) {
    return 0;
}
```

Fig. 3 Create a new class

Step 4: Enter class name and select "Create Constructor" checkbox as shown in Figure 4.



Name: Example
Arguments:
[✓] Create Constructor [ ] Create Destructor
[ ] Inherit from another class
Inheritance
Access scope: public
Inherit from class:
Header filename:
Source filename: Fall\Advanced Programming\Example\Example.cpp
Header filename: 9 Fall\Advanced Programming\Example\Example.h
[✓] Add to current project

Fig. 4 Include the class to the project

QUESTIONS

Write a class TransRot with the following capabilities:

a) Define **six private double** (i.e x, y, z, newX, newY, newZ) and a **string** data member (i.e opType).
b) The constructor of the class receives nothing. In constructor, initialize double data members with 0.0 value. **Use setXYZ function to initialize x, y, and z data members.**
c) Include *setXYZ* **public** member function which receives x, y and z coordinates and returns nothing (i.e **void** setXYZ(**double** x, **double** y, **double** z)). **In the function, use setX, setY, and setZ functions.**

d) Include *setX* **public** member function which receives x coordinate and returns nothing (i.e **void** setX (**double** x)).
e) Include *setY* **public** member function which receives y coordinate and returns nothing (i.e **void** setY (**double** y)).
f) Include *setZ* **public** member function which receives z coordinate and returns nothing (i.e **void** setZ (**double** z)).
g) Include *getX* **public** member function which receives nothing and returns x coordinate (i.e **double** getX (**void**)).
h) Include *getY* **public** member function which receives nothing and returns y coordinate (i.e **double** getY (**void**)).
i) Include *getZ* **public** member function which receives nothing and returns z coordinate (i.e **double** getZ (**void**)).
j) Include *translateAlongX* **public** member function which receives a shifting value and returns nothing (i.e **void** translateAlongX (**double** value)). In the function, calculate new position of the point according to the input value after translation operation is performed and assign the coordinates of the new position to newX, newY, and newZ data members. Also, assign the type of the operation to opType data member.
k) Repeat j) for Y and Z axes.
l) Include *rotateAroundX* **public** member function which receives an angle value in terms of degree and returns nothing (i.e **void** rotateAroundX (**double** value)). Rotation matrices are given in Figure 5.

$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$	$\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Rotation matrix by θ about x-axis	Rotation matrix by θ about y-axis	Rotation matrix by θ about z-axis

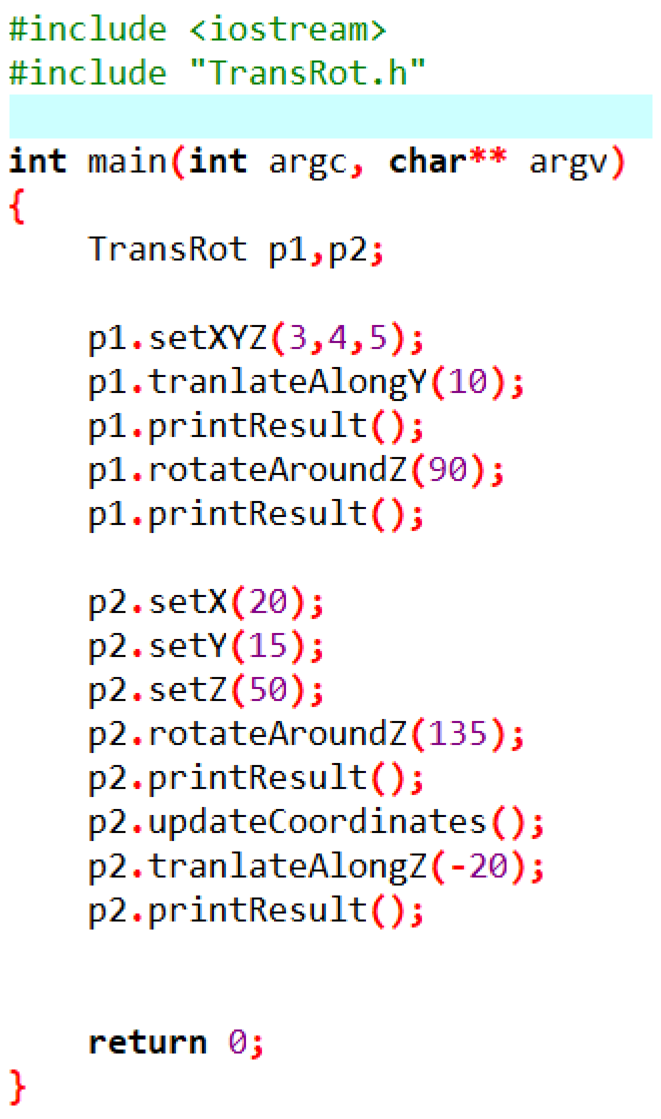
Fig. 5 Rotation Matrices [1]

In the function, calculate new position of the point according to the angle after rotation operation is performed and assign the coordinates of the new position to newX, newY, and newZ data members. Also, assign the type of the operation to opType data member. Use **getRadianFromDegree** function to convert angle from degree to radian. (**Hint: Include cmath library to use cos and sin functions.**)

m) Repeat l) for Y and Z axes.
n) Include *printResult* **public** member function which receives nothing and returns nothing (i.e **void** printResult (**void**)). In the function, print the result as in format given in Figure 7. **Use getX, getY, getZ, and getOperationType functions in the function.**
o) Include *getOperationType* **public** member function which receives nothing and returns the type of the operation (i.e **string** getOperationType (**void**)).
p) Include *getRadianFromDegree* **public** member function which receives an angle in terms of degree and returns it in terms of radian (i.e **double** getRadianFromDegree (**double value**)).
q) Include *updateCoordinates* **public** member function which receives nothing and returns nothing (i.e **void** updateCoordinates (**void**)). In the function, update x, y, and z coordinates by assigning new point's coordinates.

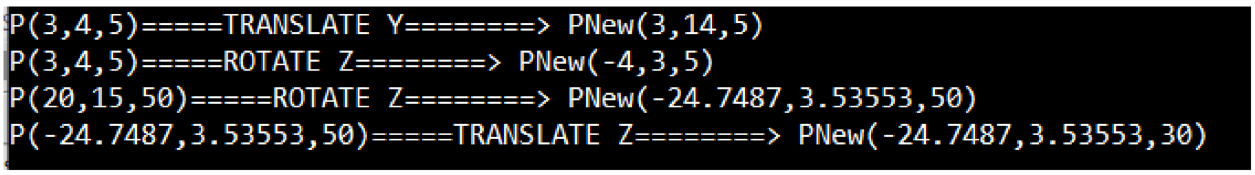
Draw UML Class Diagram for resulting class by using Visio.

Test your program with following driver program.



```cpp
#include <iostream>
#include "TransRot.h"

int main(int argc, char** argv)
{
    TransRot p1,p2;

    p1.setXYZ(3,4,5);
    p1.tranlateAlongY(10);
    p1.printResult();
    p1.rotateAroundZ(90);
    p1.printResult();

    p2.setX(20);
    p2.setY(15);
    p2.setZ(50);
    p2.rotateAroundZ(135);
    p2.printResult();
    p2.updateCoordinates();
    p2.tranlateAlongZ(-20);
    p2.printResult();


    return 0;
}
```

Fig. 6 Driver program for TransRot Class



```
P(3,4,5)=====TRANSLATE Y========> PNew(3,14,5)
P(3,4,5)=====ROTATE Z========> PNew(-4,3,5)
P(20,15,50)=====ROTATE Z========> PNew(-24.7487,3.53553,50)
P(-24.7487,3.53553,50)=====TRANSLATE Z========> PNew(-24.7487,3.53553,30)
```

Fig. 7 Terminal Screen Output for Figure 6.

[1] https://en.wikipedia.org/wiki/Rotation_matrix, October 2020.