

Creative Software Programming Practice (week-11-1)

Every assignment will be announced on **Thursday** and should be submitted by next **Tuesday**.

We have practice classes on Wednesdays and Thursdays.

The contents of the practice class are different from the assignments and aim to be completed on the same day.

Assignments will be publish on Thursday and must be submitted by the next Tuesday.

In this week **Handed out will be Nov 12, 2020, Due Nov 17, 2020**

Topics

1. Practices-1
2. Practices-2
3. Practices-3

1. Practices-1

Implement the following class Number, Square, and Cube as directed in the comments

- week-11
 - p1
 - main.cc
 - number.h
 - number.cc
 - CMakeLists.txt

```
// p1/number.h

class Number {
protected:
    int _num;
public:
    void setNumber(int num) {
        _num = num;
    }
    int getNumber() {
        return _num;
    }
};

class Square: public Number {
public:
    int getSquare(); // Implemented to return the square of the number specified
    by setNumber()
};

class Cube: public Square {
public:
```

```
int getCube (); // Implemented to return the cube of the number specified by
setNumber()
};
```

This program should take user input repeatedly

- 'number' [number] - Create a Number object and print out the return value of getNumber() as shown in the following example.
- 'square' [number] - Create a Square object and print out the return value of getNumber() and getSquare() as shown in the following example.
- 'cube' [number] - Create a Cube object and print out the return value of getNumber(), getSquare(), and getCube() as shown in the following example.
- 'quit' – Quit the program.

Input:

```
number 3
square 2
cube 4
quit
```

Output:

```
getNumber(): 3
getNumber(): 2
getSquare(): 4
getNumber(): 4
getSquare(): 16
getCube(): 64
```

Pratice-2

The following program prints out 2. Modify the program using one of the C ++ type casting operators only once so that it prints out 2.5 (actual division result).

- week-11
 - p2
 - main.cc
 - CMakeLists.txt

```
// p2/main.cc
#include <iostream>

int main() {
    int a = 10;
    int b = 4;
    cout << a / b << endl;
    return 0;
}
```

Input:

(No input)

Output:

2.5

Pratice-3

Write a program that works as follows using the following code

- week-11
 - p3
 - main.cc
 - CMakeLists.txt
- Take arbitrary number of "B", "C", and "D" strings, and then creates one object of class B, C, or D for each string, and put those object into `std::vector<B*> arr`.
- When the user enters 0, the for statement traverses each element of arr,
 - Call `C::test_C()` if the element is a C type object
 - Call `D::test_D()` if the element is a D type object (using `dynamic_cast`)
 - Do nothing for the B type object
- Note that this problem is intended to practice how to use `dynamic_cast`, so remember that using `dynamic_cast` in this way is not desirable.
- Each element of arr must be deallocated after use.
- This program should take user input repeatedly

```
class B {
public:
    virtual ~B() {}
};

class C : public B {
public:
    void test_C() { std::cout << "C::test_C()" << std::endl; }
};

class D : public B {
public:
    void test_D() { std::cout << "D::test_D()" << std::endl; }
};
```

Input:

B
C
D
B
C
0

Output:

```
C::test_C()  
D::test_D()  
C::test_C()
```