1.1

Frame

```
weicaivi@linux6:~/mpcs51044-cpp/build$ /home/weicaivi/mpcs51044-cpp/build/frame
What's your name? Wei

**********************
*                    *
*                    *
*                    *
*                    *
*                    *
*     Hello, Wei!    *
*                    *
*                    *
*                    *
*                    *
*                    *
**********************
```

Frame2

```
weicaivi@linux6:~/mpcs51044-cpp/build$ /home/weicaivi/mpcs51044-cpp/build/frame2
What's your name? Wei

***************
*             *
* Hello, Wei! *
*             *
***************
```

personalizedHello

```
weicaivi@linux6:~/mpcs51044-cpp/build$ /home/weicaivi/mpcs51044-cpp/build/persona
lized_hello
What's your name? Wei
Hello, Wei!
```

vector_simple_demo

```
weicaivi@linux6:~/mpcs51044-cpp/build$ /home/weicaivi/mpcs51044-cpp/build/vector_simple_demo
1, 4, 16, weicaivi@linux6:~/mpcs51044-cpp/build$
```

1.2 Pascal's triangle

```
weicaivi@linux6:~/mpcs51044-cpp/build$ /home/weicaivi/mpcs51044-cpp/build/pascals_triangle
                1
              1   1
            1   2   1
          1   3   3   1
        1   4   6   4   1
      1   5  10  10   5   1
    1   6  15  20  15   6   1
  1   7  21  35  35  21   7   1
```

1.4 valid C program but invalid C++ program

```
weicaivi@linux6:~/mpcs51044-cpp/hw1/1.4$ gcc valid.c -o valid
weicaivi@linux6:~/mpcs51044-cpp/hw1/1.4$ ./valid
Class number: 10
weicaivi@linux6:~/mpcs51044-cpp/hw1/1.4$ g++ valid.c -o valid
valid.c: In function ‘int main()’:
valid.c:4:5: error: expected primary-expression before ‘int’
    4 |     int class = 10;  // 'class' is a valid identifier in C but a reserved keyword in C++
      |     ^~~
valid.c:5:34: error: expected primary-expression before ‘class’
    5 |     printf("Class number: %d\n", class);
      |                                  ^~~~~
```

1.5

C++ is called "C++" as a play on the increment operator (++), which is used in C (and later in C++) to increment a variable's value by one. This naming was chosen to symbolize the language's evolution as an "increment" or enhancement of the C programming language. C++ extends C by adding object-oriented features, among other improvements, effectively making it "C incremented."

The postfix increment operator "++" increments the value of the variable but returns the original value before it was incremented. This is somewhat analogous to how C++ was developed: it builds upon and extends the original value (capabilities) of C.

If it were named "++C" using the prefix increment logic, it implies that the change (increment) happens first before using the value. This might suggest a more radical departure or redefinition from C, which isn't the case since C++ is built as an extension and largely maintains backward compatibility with C. Therefore, "C++" suggests enhancing and building upon existing foundations, which is the core philosophy behind the development of C++.