Introduction to C++ Classes:

You can find an example implementation of Time class below using C++ Classes.

Time.h

```
#ifndef TIME_H
#define TIME_H

class Time {
private:
        int hour;
        int minute;
        int second;
public:
        Time();
        void setTime(int, int, int);
        void printTime(void);
};
#endif
```

Time.cpp

```
#include <iostream>
#include "Time.h"

using std::cout;

Time::Time() {
        hour = 0;
        minute = 0;
        second = 0;
}

void Time::setTime(int h, int m, int s) {
        hour = h;
        minute = m;
        second = s;
}

void Time::printTime(void){
        cout << ((hour < 10) ? "0" : "") << hour << ":";
        cout << ((minute < 10) ? "0" : "") << minute << ":";
        cout << ((second < 10) ? "0" : "") << second << "\n";
}
```

main.cpp

```
#include <iostream>
#include "time.h"
using namespace std;

int main()
{
	Time mytime1;
	mytime1.setTime(13, 15, 10);
	mytime1.printTime();

	Time *mytime2 = new Time();
	mytime2->setTime(5, 13, 45);
	mytime2->printTime();

	delete mytime2;

	return 0;
}
```

Output

```
13:15:10
05:13:45
```

Practical Exercises on C++ Classes

a. Create a class Rectangle. The class has attributes length and width, each of which defaults to 1. It has member functions that calculate the perimeter and the area of the rectangle. It has set and get functions for both length and width. The set function should verify that length and width are each floating point numbers larger than 0.0 and lower than 20.0

b. Modify the Rectangle class implemented in Part (a) to include draw function that displays the rectangle. However, you first need to include the following member functions:

- setFillCharacter function to specify the character (such as -) that will be used to fill the body of the rectangle.
- setPerimeterCharacter function to specify the character (such as *) that will be used to draw the border of the rectangle.

It can display the integer portions of the given length and width, for example following output can be drawn for any rectangle with length between 5 to 6 (less than 6) and width between 15 to 16(less than 16).

```
***************
*-------------*
*-------------*
*-------------*
***************
```

c. Implement a Coordinate class to hold (x,y) values. Create another class Line which consists of start and end coordinates, and line type. The default values should be as follows: start coordinate is (0, 0), end coordinate is (0, 0). This class should support two types of lines which are vertical (v) and horizontal (h). The default value for line type should be vertical. This class has set function which is used to set the start and end coordinates, and line type. This function should verify that the line can be drawn using the start and end coordinates based on the line type. This class also has draw function which draws a line.

Draw function output

Horizontal Line Example
start: (5,3), end: (12,3), horizontal
Output:

```
3 > * * * * * * * *
    ^           ^
    5           12
```

Vertical Line Example
start: (5,3), end: (5,10), vertical
Output:

```
* < 3
*
*
*
*
*
*
* < 10
^
5
```

Invalid Line Example 1
start: (5,3), end: (12,3), vertical

Output:
Cannot be drawn

Invalid Line Example 2
start: (6,4), end: (7,2), horizontal

Output:
Cannot be drawn

References:

1. H. M. Deitel and P. J. Deitel (2000) C How to Program. Third Edition. Prentice Hall: New Jersey