

C++ Program Design/3e
Chapter 7 – The Class Construct and Object Oriented Design
Answers to Self-Check Exercises

1. answer: Inspector.
2. answer: Mutator.
3. answer: Facilitator.
4. answer: Constructor.
5. answer: Instantiation.
6. answer: `BigBox.SetColor(Green);`
7. answer: `Button.SetSize(4.0, 3.0);`
8. answer: `Door.SetPosition(4.0, 2.0);`
9. answer: The private data member `PhoneNumber` cannot be accessed.
10. answer: Private by default, since no access label precedes its declaration.
11. answer:

```
class Date {
public:
    Date(int year = 1, int month = 1, int day = 1);
    int GetYear() const;
    int GetMonth() const;
    int GetDay() const;
    void SetYear(int year);
    void SetMonth(int month);
    void SetDay(int day);
private:
    int Year;
    int Month;
    int Day;
};
```
12. answer: This is not a question; it is actually a part of the previous question, # 11.

13. answer:

```
class Book {
public:
    Book(string Title = "No title",
         string Author "No author", int ID = 0);
    string GetTitle() const;
    string GetAuthor () const;
    int GetID() const;
    void SetTitle(string title);
    void SetAuthor(string author);
    void SetID(int id);
private:
    string Title;
    string Author;
    int ID;
};
```

14. answer:

```
enum Direction {North, South};

class Latitude {
public:
    Latitude(int Degrees = 0, int Minutes = 0,
            int Seconds = 0, Direction = North);
    int GetDegrees() const;
    int GetMinutes() const;
    int GetSeconds() const;
    Direction GetDirection() const;
    void SetDegrees(int degrees);
    void SetMinutes(int minutes);
    void SetSeconds(int seconds);
    void SetDirection(Direction dir);
private:
    int Degrees;
    int Minutes;
    int Seconds;
    Direction NorthOrSouth;
};
```

15. answer:

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

using namespace std;

SimpleWindow ColorWindow("Color Palette", 5.0, 8.0);

int ApiMain() {
    const float StripeWidth = 5.0;
    const float StripeHeight = 1.0;
    const float XPosition = 2.5;
    float YPosition = 0.5;

    ColorWindow.Open();

    RectangleShape ColorPatch(ColorWindow, XPosition,
        YPosition, White, StripeWidth, StripeHeight);

    for (color c = Black; c <= Magenta; c = (color)(c+1)) {
        ColorPatch.SetColor(c);
        ColorPatch.SetPosition(XPosition, YPosition);
        ColorPatch.Draw();
        YPosition += StripeHeight;
    }

    return 0;
}
```