Part I – Basics

1. (8) What is a function **prototype** and what purpose does it serve?

2. (15) Define the terms **class** and **object** as they are used in C++. Clearly distinguish between them.

3. (7) How are the components of a class that can be accessed outside the class distinguished from those that can be directly accessed only within the class? Which class components are normally coded in each of these sections?

4. (7) A pre- and post-condition are often included as comments at the beginning of a function. What is the purpose of these comments? What information do they specify?

Part II – Application of Principles

1. (10) Describe briefly the process by which the source code for a C++ program is transformed into an executable program. Be sure to mention the role of each of the following: source code, header files, object code, library files, executable program. (You may answer in words or with a diagram.)

2. (5) Consider the code fragment

   ```cpp
   for (int i = 0; i <= 10; i++)
     cout << i << " squared = " << i*i << endl;
   ```

   Write equivalent code using a **while** or **do-while** statement instead of **for**.
Part III – Programming
In this section, you are to use library functions and classes described in the text wherever possible. Employ your best programming practice. Write complete programs but do not include comments.

**prompt.h summary**
- int PromptRange(const string & prompt, int low, int high)
- const string PromptString(const string & prompt)
- bool PromptYesNo(const string & prompt)

3. (15) Write code that prompts the user for a number of fence posts in a fence and then draws a fence as shown below.

   **Sample Program Execution**
   Enter the number of fence posts: 8
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

4. (19) Consider the class Game described below. It is required to add a method Autoplay that takes one integer parameter. When this new method is called, it will play the specified number of games, betting $10 on each game, unless the bankroll is reduced to zero in which case play will stop. At the end of play, the method will print a message of the form “After playing nn games, I have $ddd..”. Indicate the changes that need to be made in the class declaration below and code the implementation of Autoplay. The implementation is simple if you take advantage of the methods already coded.

```cpp
class Game {
    public:
        Game();
        bool Continue();
        void MakeBet();
        void Play();
    private:
        int myBankRoll;
        int myWager;
        bool WinOne();
        bool GetPoint(int point);
        int RollTwo();
};
```

5. (14) Write a main program that uses the Game class as modified in problem 8, above, to prompt the user for a number of games to play and then plays the specified number of games. This should be a complete program file that could be compiled and linked with the files game.h and game.cc defined in problem 8.