

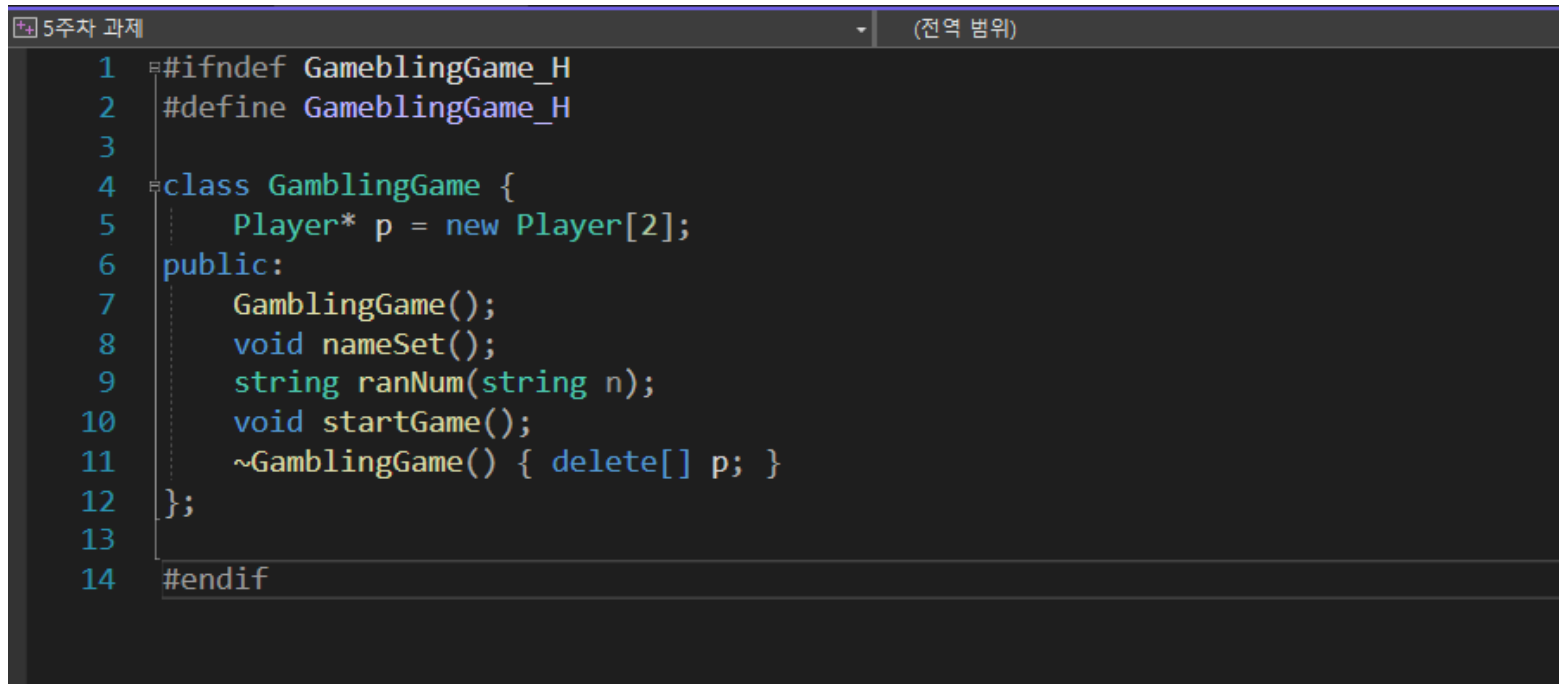
Player.h

```
5주차 과제 (전역 범위)
1  #ifndef Player_H
2  #define Player_H
3
4  class Player {
5      string name;
6  public:
7      void setName(string name);
8      string getName() { return name; };
9  };
10
11 #endif
```

1, 2, 11번 라인의 `#ifndef`, `#define`, `#endif` 명령어를 통하여 `Player_H`가 이미 정의 되어있다면 중복으로 정의되는것을 방지합니다.

4~9번 라인을 통하여 `Player`클래스를 정의하고 클래스 내부에 이름을 설정할 수 있는 `setName` 함수와 저장된 이름을 반환하는 `getName`함수를 정의합니다.

GamblingGame.h



```
5주차 과제 (전역 범위)
1  #ifndef GamblingGame_H
2  #define GamblingGame_H
3
4  class GamblingGame {
5      Player* p = new Player[2];
6  public:
7      GamblingGame();
8      void nameSet();
9      string ranNum(string n);
10     void startGame();
11     ~GamblingGame() { delete[] p; }
12 };
13
14 #endif
```

1, 2, 14번 라인의 `#ifndef`, `#define`, `#endif` 명령어를 통하여 `GamblingGame_H`가 이미 정의 되어있다면 중복으로 정의되는것을 방지합니다.

4~12번 라인을 통하여 `GamblingGame`클래스를 정의하여 2명의 플레이어 배열을 포인터를 이용하여 동적으로 선언합니다.

7번 라인에 `GamblingGame`의 생성자를 생성, 8~10번 라인에 플레이어 이름 설정 함수, 난수 생성 및 게임 결과 반환 함수, 게임 시작 함수를 각각 선언하고, 11번 라인에서 소멸자`GamblingGame`을 이용하여 동적으로 할당된 플레이어의 메모리를 해제합니다.

GamblingGame.cpp

```
5주차 과제
GamblingGame

1 #include<iostream>
2 #include<string>
3 #include <cstdlib>
4 #include <ctime>
5 using namespace std;
6
7 #include "Player.h"
8 #include "GamblingGame.h"
9
10 GamblingGame::GamblingGame() {
11     cout << "***** 갠블링 게임을 시작합니다. *****\n";
12     srand(time(NULL));
13 }
14
15 void GamblingGame::nameSet() {
16     string name;
17     cout << "첫번째 선수 이름>>";
18     cin >> name;
19     p[0].setName(name);
20     cout << "두번째 선수 이름>>";
21     cin >> name;
22     p[1].setName(name);
23 }
24
25 string GamblingGame::ranNum(string n) {
26     int r[3];
27     cout << "\t\t";
28     for (int i = 0; i < 3; i++) {
29         r[i] = rand() % 3;
30         cout << r[i] << "\t";
31     }
32     if (r[0] == r[1] && r[0] == r[2]) {
33         n += "님 승리!!";
34         return n;
35     }
36     else
37         return "아쉽군요!";
38 }
39 }
```

7~8번 라인을 통하여 Player.h, GamblingGame.h 헤더파일을 include합니다.

10~13번 라인에 생성자가 호출되면 "도박 게임을 시작합니다"라는 메시지를 출력하고, srand(time(NULL))을 이용하여 난수 생성의 시드를 현재 시간을 기준으로 설정하여 매번 다른 난수를 생성할수 있도록 합니다.

15~23번 라인을 통하여 player의 이름을 입력받아 setName함수를 통하여 p배열에 플레이어의 이름을 설정합니다.

25~38번 라인을 통하여 rand함수를 통해 1~3까지의 난수를 저장하는 r배열을 생성하여 3개의 난수를 저장하고 3개의 난수가 모두 같을 경우 승리를 같지 않을 경우 아쉽군요를 출력하도록 만듭니다.

GamblingGame.cpp

```
39
40 void GamblingGame::startGame() {
41     string n;
42     int i = 0;
43     while (true) {
44         string m;
45         cout << p[i % 2].getName() << ":\n";
46         getline(cin, n);
47         m = p[i % 2].getName();
48         n = ranNum(n);
49         if (n == "님 승리!!") {
50             cout << m + n;
51             break;
52         }
53         else
54             cout << n << endl;
55         i++;
56     }
57 }
58
59 void Player::setName(string n) {
60     name = n;
61 }
```

40~57번 라인에 각 플레이어의 이름을 기반으로 ranNum() 함수를 호출해 while 문을 통하여 반복으로 게임을 진행하고 결과를 확인합니다.

또한 if문을 이용하여 한명이 승리할시 반복문을 멈출수 있도록 만듭니다.

59~61번 라인을 이용하여 name = n으로 설정합니다.

main.cpp

```
5주차 과제 (전역 범위)
1 #include<iostream>
2 #include<string>
3 #include <cstdlib>
4 #include <ctime>
5 using namespace std;
6
7 #include "Player.h"
8 #include "GamblingGame.h"
9
10 int main() {
11     GamblingGame game;
12     game.nameSet();
13     game.startGame();
14 }
```

7~8번 라인을 통하여 Player.h, GamblingGame.h 헤더파일을 include합니다.
10~14번 라인에서 메인함수에서 GamblingGame객체를 생성하고 nameSet(), startGame()을 불러와서 플레이어의 이름 설정 및 게임을 진행합니다.