

## 1. GamblingGame.h 헤더파일

```
#include <iostream>

#include <string>

using namespace std;

class GamblingGame {
public:
    int getRandom();
};

class Player {
public:
    void setName(string);
    string getName();
    string name;
};
```

## 2. GamblingGame.cpp

```
#include <iostream>

#include <string>

#include "GamblingGame.h"

using namespace std;

void Player::setName (string playerName) {
    name = playerName;
}

string Player::getName()
{
    return name;
}

int GamblingGame::getRandom() {
    int num;

    num = rand() % 3;

    return num;
}
```

### 3. Main.cpp 메인함수

```
#include <iostream>

#include <string>

#include "GamblingGame.h"

using namespace std;

int main() {

    Player player[2];

    GamblingGame g[3];

    string name;

    int count = 0;

    int ranNum[3];

    srand((unsigned int)time(0));

    cout << "***** 겜블링 게임을 시작합니다. *****" << endl;

    cout << "첫번째 선수 이름 >> ";

    cin >> name;

    player[0].setName(name);

    cout << "두번째 선수 이름 >> ";

    cin >> name;

    player[1].setName(name);

    for (;;)

    {
```

```

cin.ignore();

cout << player[count].getName() << ": <Enter>";

cin.get();

for (int i = 0; i < 3; i++)
{
    ranNum[i] = g[i].getRandom();
}

if (ranNum[0] == ranNum[1] && ranNum[1] == ranNum[2])
{
    cout << "                " << ranNum[0] << "                " <<
ranNum[1] << "                " << ranNum[2] << "                "
<< player[count].getName() << "님 승리!!" << endl;

    break;
}

else
{
    cout << "                " << ranNum[0] << "                " <<
ranNum[1] << "                " << ranNum[2] << "                "
<< "아쉽군요!";
}

count = (count + 1) % 2;
}

}

```

1. 겜블링 게임은 플레이어 2명이 번갈아가며 세개의 무작위 숫자가 동일할때 승리하는 게임이다.
2. 이 게임을 구현하기 위해서는 플레이어의 이름을 저장할 수 있는 name변수와 무작위 숫자를 생성하기 위해 객체의 배열을 사용했다. 무작위 숫자는 rand()를 사용했다. 반복문을 사용하여, 승리 조건을 만족하기 전까지 반복되게 작성했다.
3. Player클래스에서는 플레이어의 이름을 입력받을 배열을 선언하고, 세터를 이용해서 저장한다.

세 개의 무작위 숫자는 GamblingGame클래스에서 마찬가지로 객체 배열을 통해 선언했다. 게임의 승리조건에 만족하려면 세 숫자가 같아야 하는데, ranNum배열에 세 개의 무작위 숫자들을 담아서 비교할 수 있도록 작성했다.

즉, 승리조건을 비교하는 if문에 getRandom을 넣으면 호출할 때마다 다른 값이 반환되므로 무작위 숫자를 다른 배열에 한번 더 저장후에 비교해야 의도한 바로 실행이 된다.

그리고 세개의 무작위 숫자를 생성할 때, rand() % 3로 작성해서 0부터 2까지의 무작위 숫자를 생성하게 작성했다. 작성한대로 실행해보니 무작위 숫자이기는 하지만, 항상 결과값이 같았다는 문제가 있었다. 이를 해결하기 위해 코드에서 srand((unsigned int)time(0));

이 부분을 작성했다. srand()메서드는 초기값, 즉 시드를 설정하는데, 이 시드값을 time()메서드로 현재 시간으로 설정해서, 프로그램을 실행할때마다 매번 다른 무작위 숫자들을 생성하게 작성했다. 시드값은 음수가 나오지 않게 타입 캐스팅 (unsigned int)으로 작성했다.

엔터키를 눌러야 다음 플레이어의 무작위 숫자를 출력되도록 작성했다. cin.ignore()를 통해 마지막으로 입력했던 버퍼(엔터키 = \n)를 제거했다.

그리고 cin.get()을 사용해서 엔터키를 입력받기 전까지 대기하도록 작성했다.