Stony Brook University
College of Engineering and Applied Science

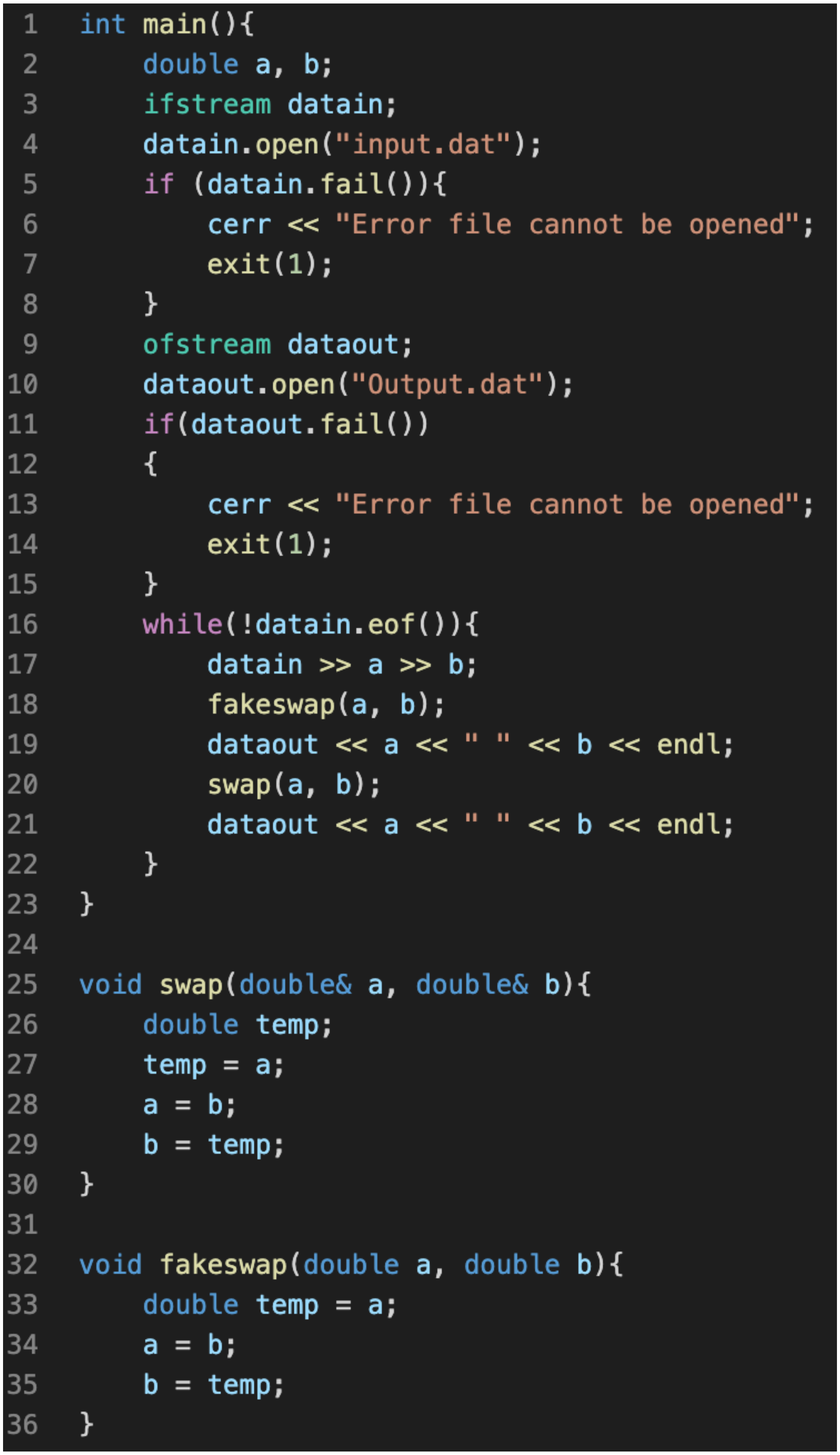
ESE 224.L02

Lab 4

Ryan Lin

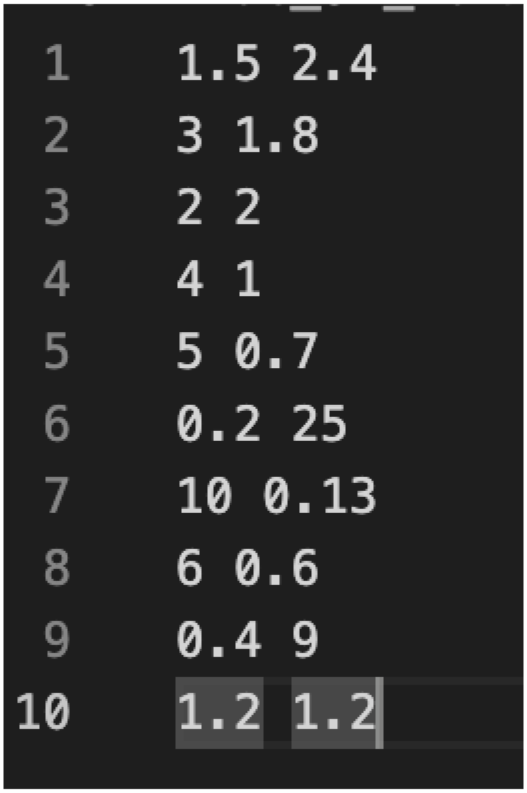
Professor: Xin Wang

Part 1:
Main.cpp



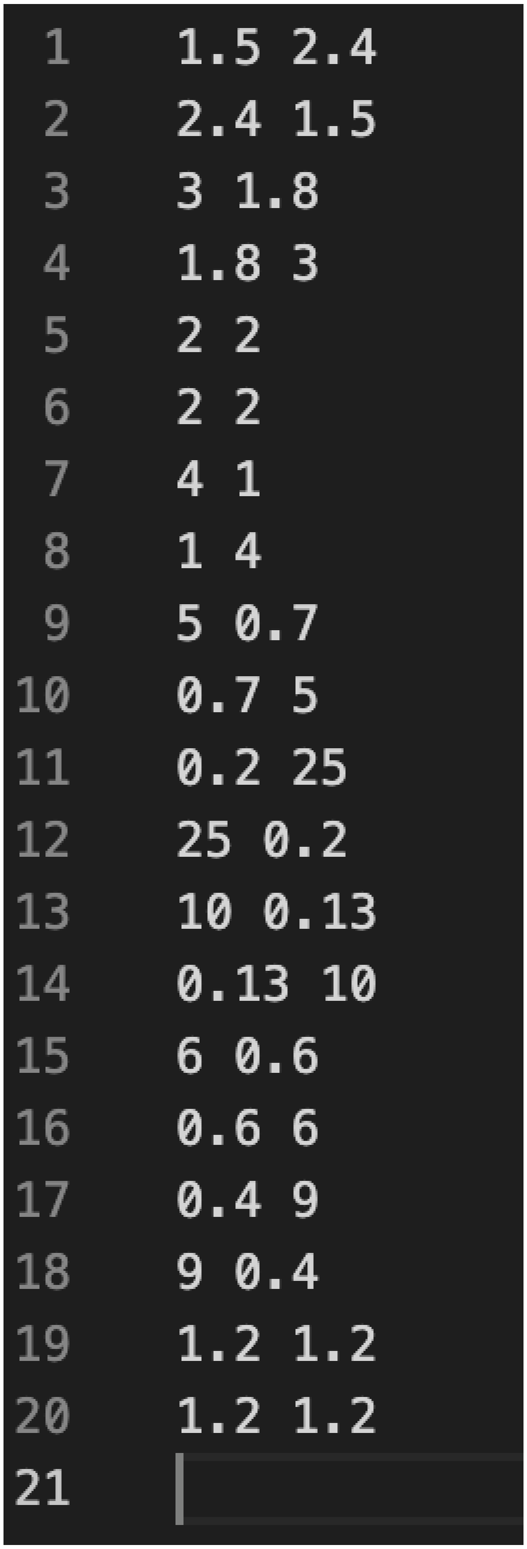
```cpp
int main(){
    double a, b;
    ifstream datain;
    datain.open("input.dat");
    if (datain.fail()){
        cerr << "Error file cannot be opened";
        exit(1);
    }
    ofstream dataout;
    dataout.open("Output.dat");
    if(dataout.fail())
    {
        cerr << "Error file cannot be opened";
        exit(1);
    }
    while(!datain.eof()){
        datain >> a >> b;
        fakeswap(a, b);
        dataout << a << " " << b << endl;
        swap(a, b);
        dataout << a << " " << b << endl;
    }
}

void swap(double& a, double& b){
    double temp;
    temp = a;
    a = b;
    b = temp;
}

void fakeswap(double a, double b){
    double temp = a;
    a = b;
    b = temp;
}
```

Input.dat



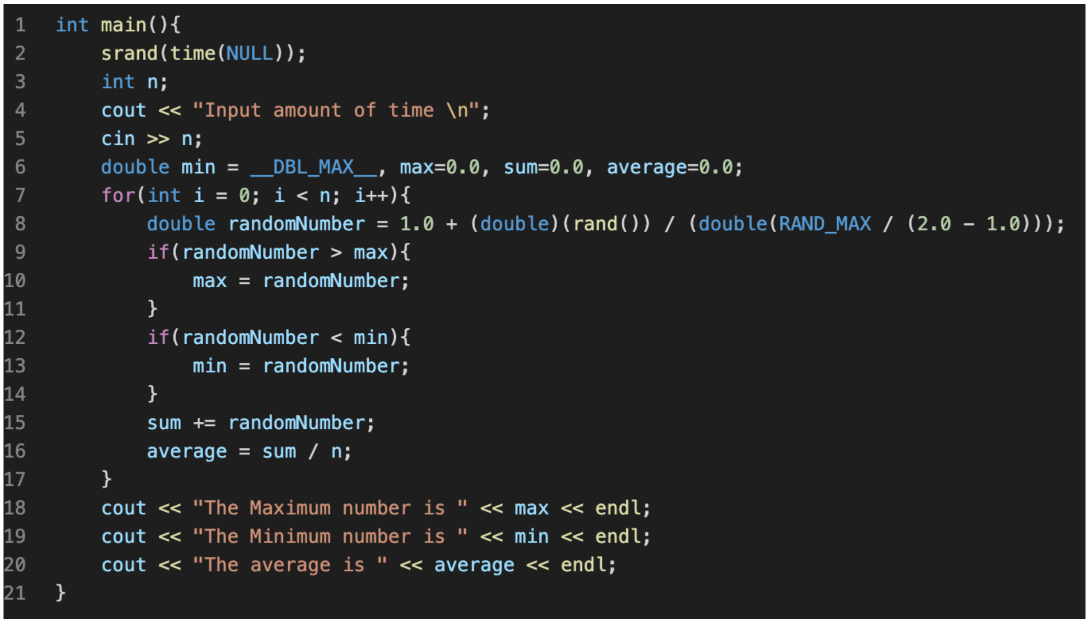
```
1.5 2.4
3 1.8
2 2
4 1
5 0.7
0.2 25
10 0.13
6 0.6
0.4 9
1.2 1.2
```

Output.dat



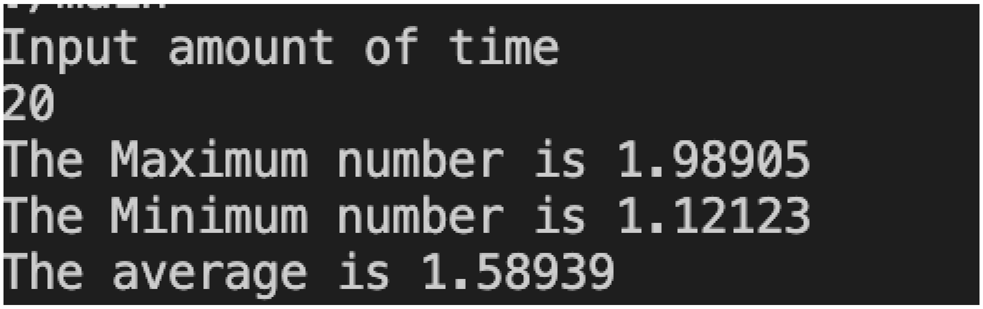
```
1.5 2.4
2.4 1.5
3 1.8
1.8 3
2 2
2 2
4 1
1 4
5 0.7
0.7 5
0.2 25
25 0.2
10 0.13
0.13 10
6 0.6
0.6 6
0.4 9
9 0.4
1.2 1.2
1.2 1.2

```

Part 2:
main.cpp



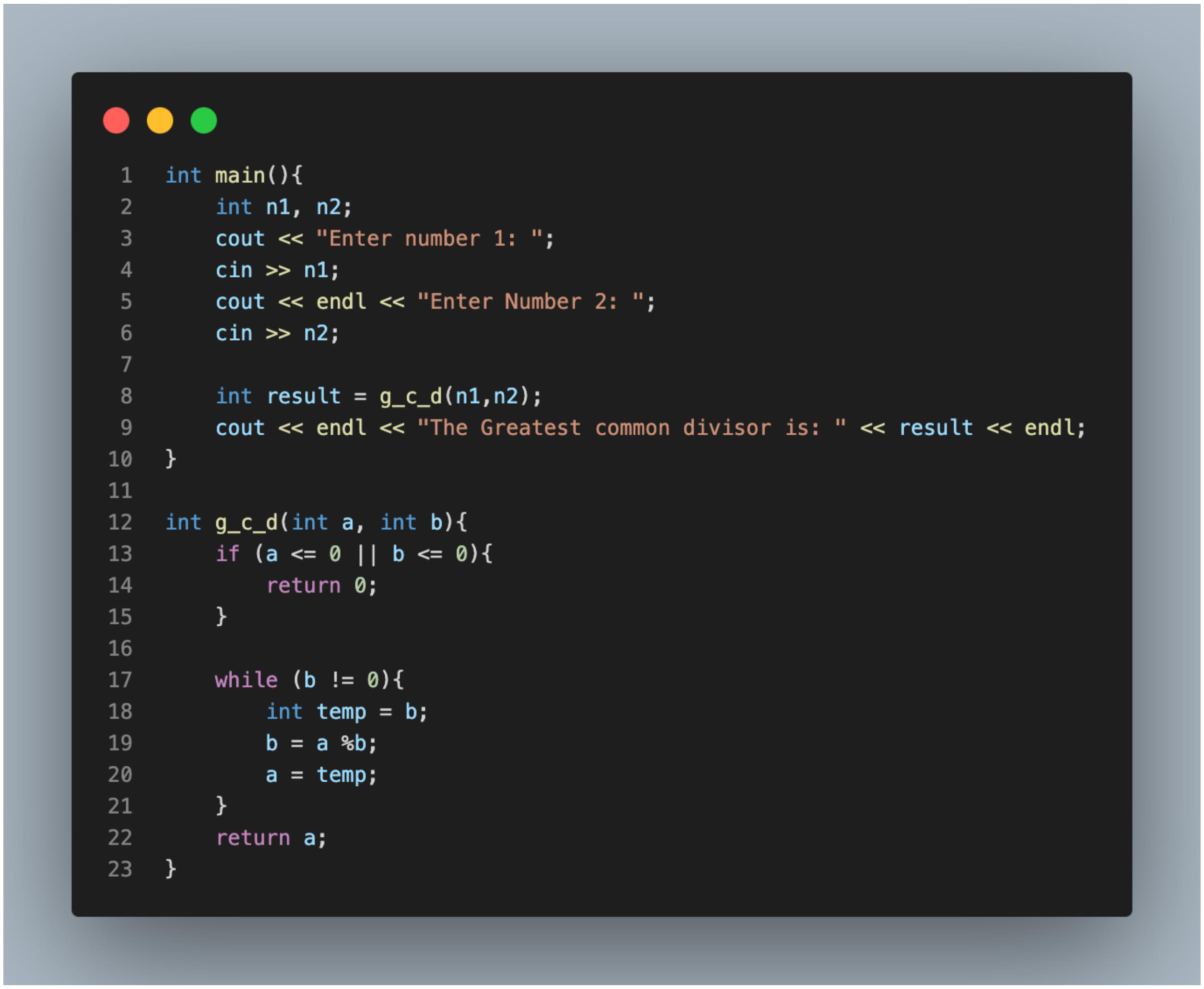
```cpp
int main(){
    srand(time(NULL));
    int n;
    cout << "Input amount of time \n";
    cin >> n;
    double min = __DBL_MAX__, max=0.0, sum=0.0, average=0.0;
    for(int i = 0; i < n; i++){
        double randomNumber = 1.0 + (double)(rand()) / (double(RAND_MAX / (2.0 - 1.0)));
        if(randomNumber > max){
            max = randomNumber;
        }
        if(randomNumber < min){
            min = randomNumber;
        }
        sum += randomNumber;
        average = sum / n;
    }
    cout << "The Maximum number is " << max << endl;
    cout << "The Minimum number is " << min << endl;
    cout << "The average is " << average << endl;
}
```

Output



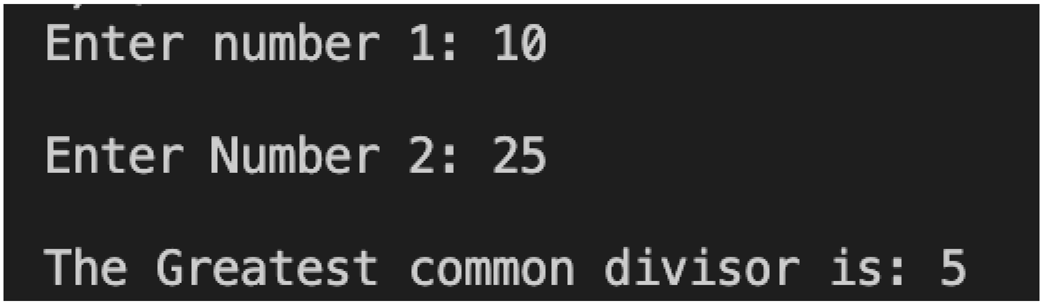
```
Input amount of time
20
The Maximum number is 1.98905
The Minimum number is 1.12123
The average is 1.58939
```

Part 3:
Main.cpp



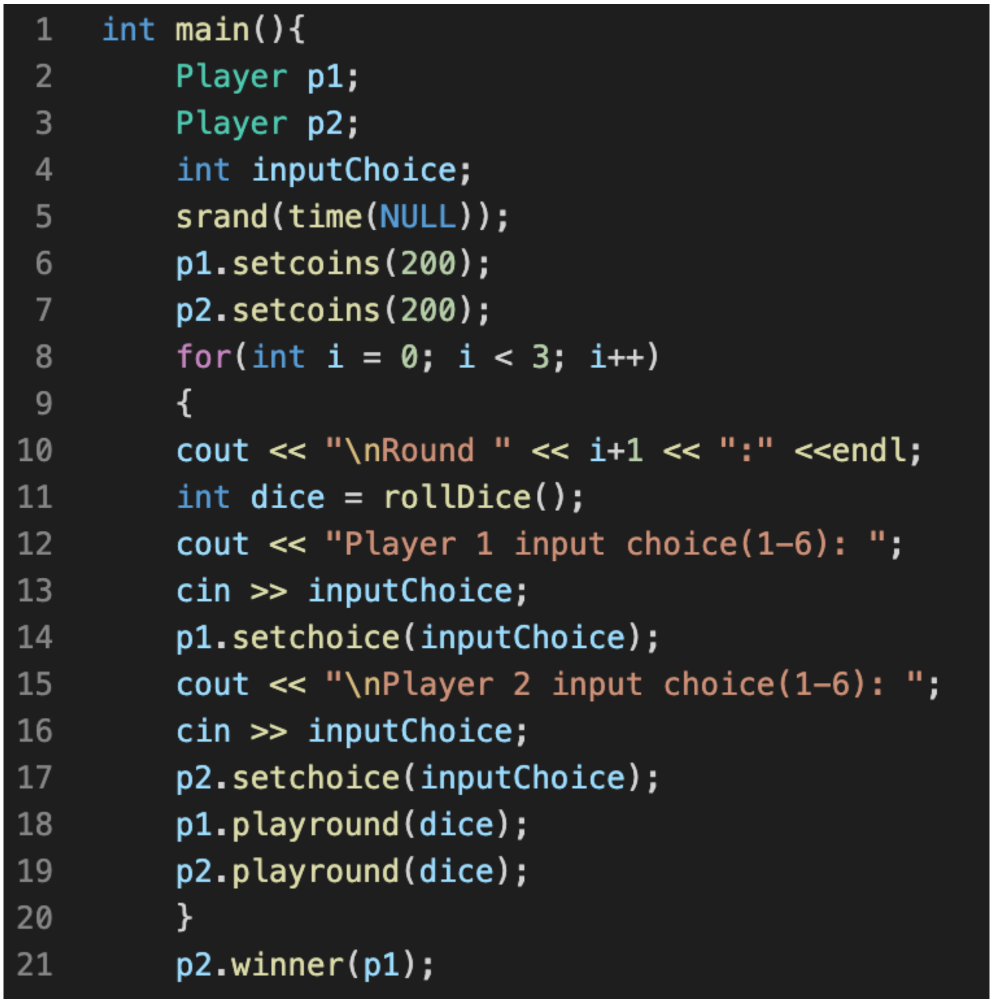
```cpp
int main(){
    int n1, n2;
    cout << "Enter number 1: ";
    cin >> n1;
    cout << endl << "Enter Number 2: ";
    cin >> n2;

    int result = g_c_d(n1,n2);
    cout << endl << "The Greatest common divisor is: " << result << endl;
}

int g_c_d(int a, int b){
    if (a <= 0 || b <= 0){
        return 0;
    }

    while (b != 0){
        int temp = b;
        b = a %b;
        a = temp;
    }
    return a;
}
```

Output



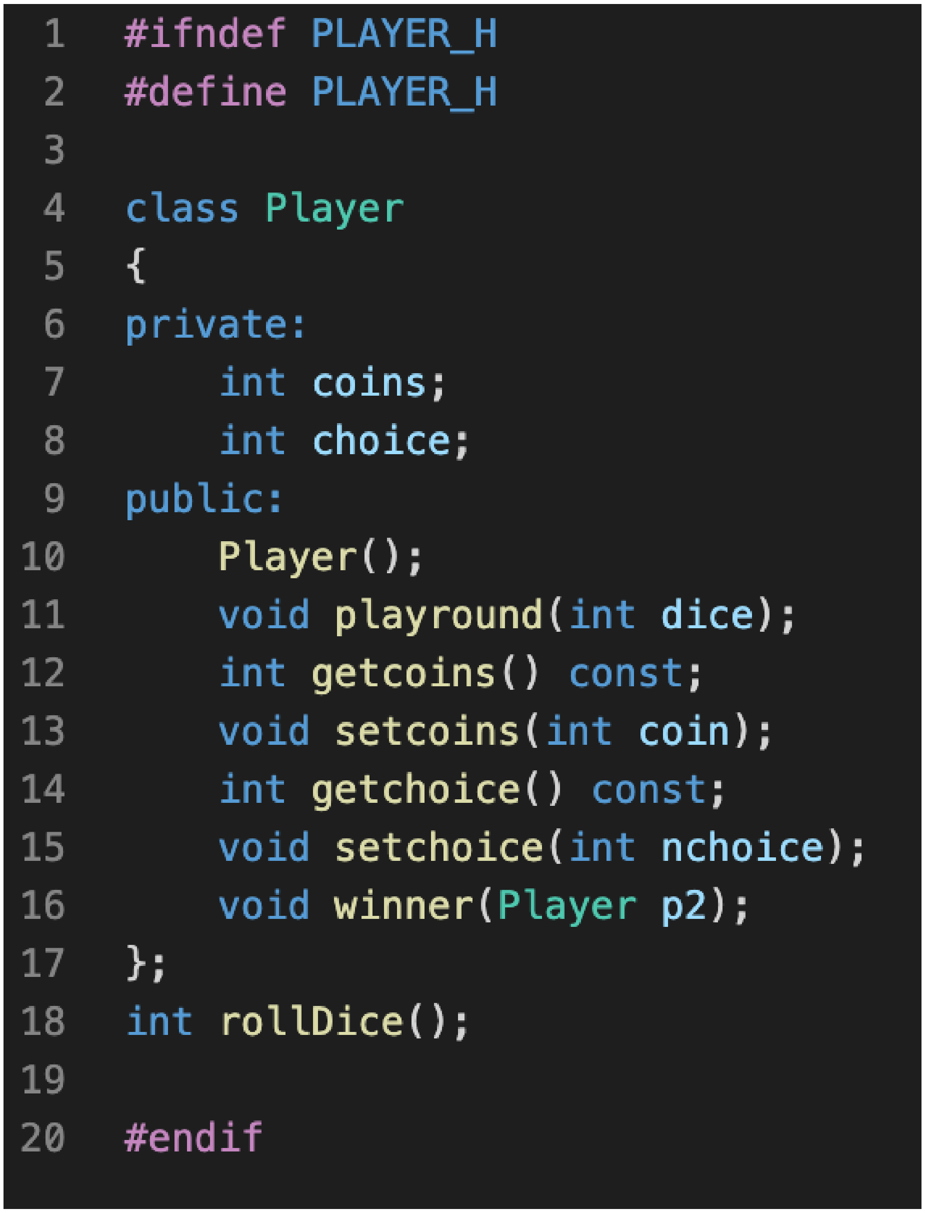
```
Enter number 1: 10

Enter Number 2: 25

The Greatest common divisor is: 5
```

Part 4:
Main.cpp



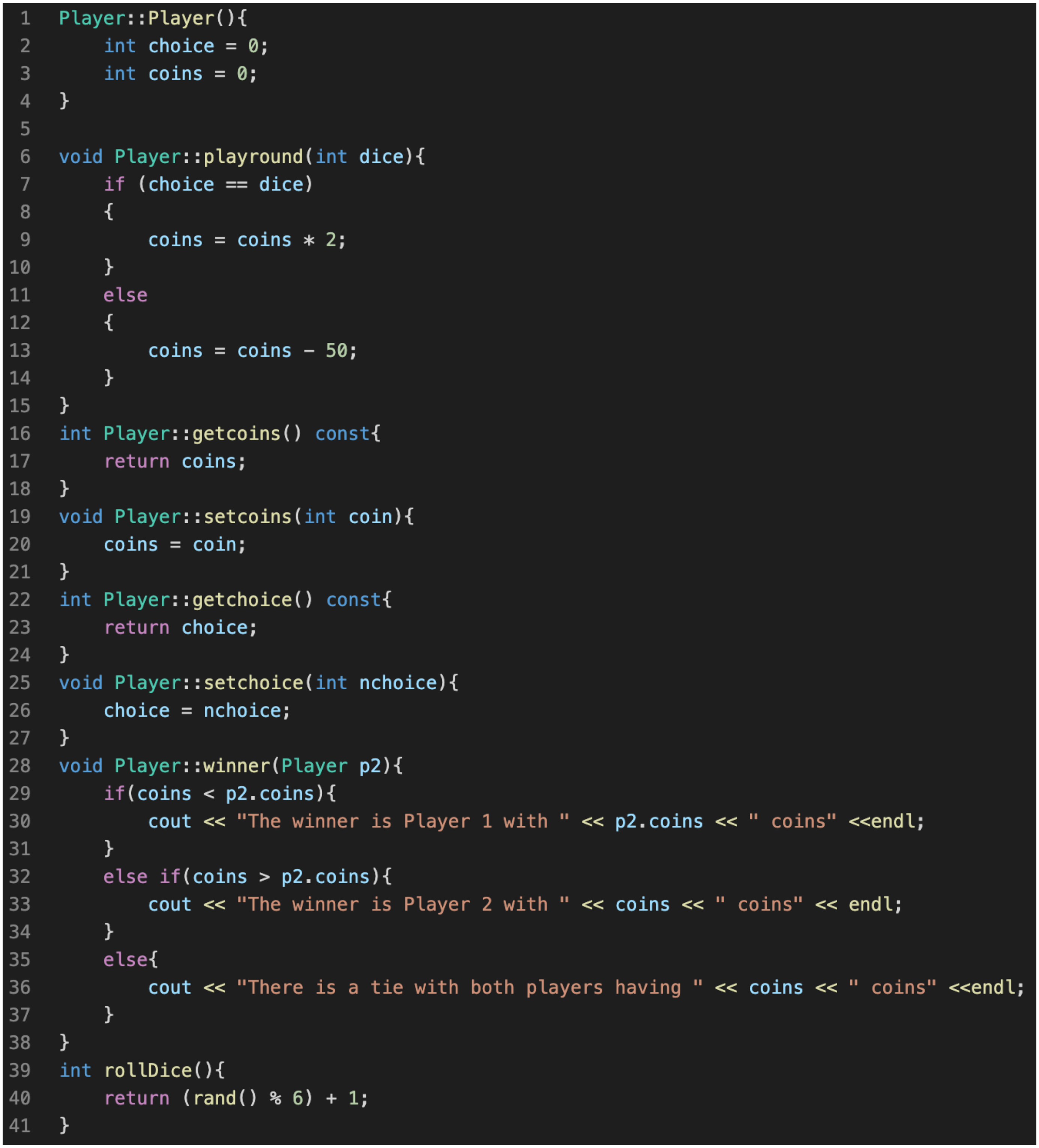
```
int main(){
    Player p1;
    Player p2;
    int inputChoice;
    srand(time(NULL));
    p1.setcoins(200);
    p2.setcoins(200);
    for(int i = 0; i < 3; i++)
    {
    cout << "\nRound " << i+1 << ":" <<endl;
    int dice = rollDice();
    cout << "Player 1 input choice(1-6): ";
    cin >> inputChoice;
    p1.setchoice(inputChoice);
    cout << "\nPlayer 2 input choice(1-6): ";
    cin >> inputChoice;
    p2.setchoice(inputChoice);
    p1.playround(dice);
    p2.playround(dice);
    }
    p2.winner(p1);
```

Player.h



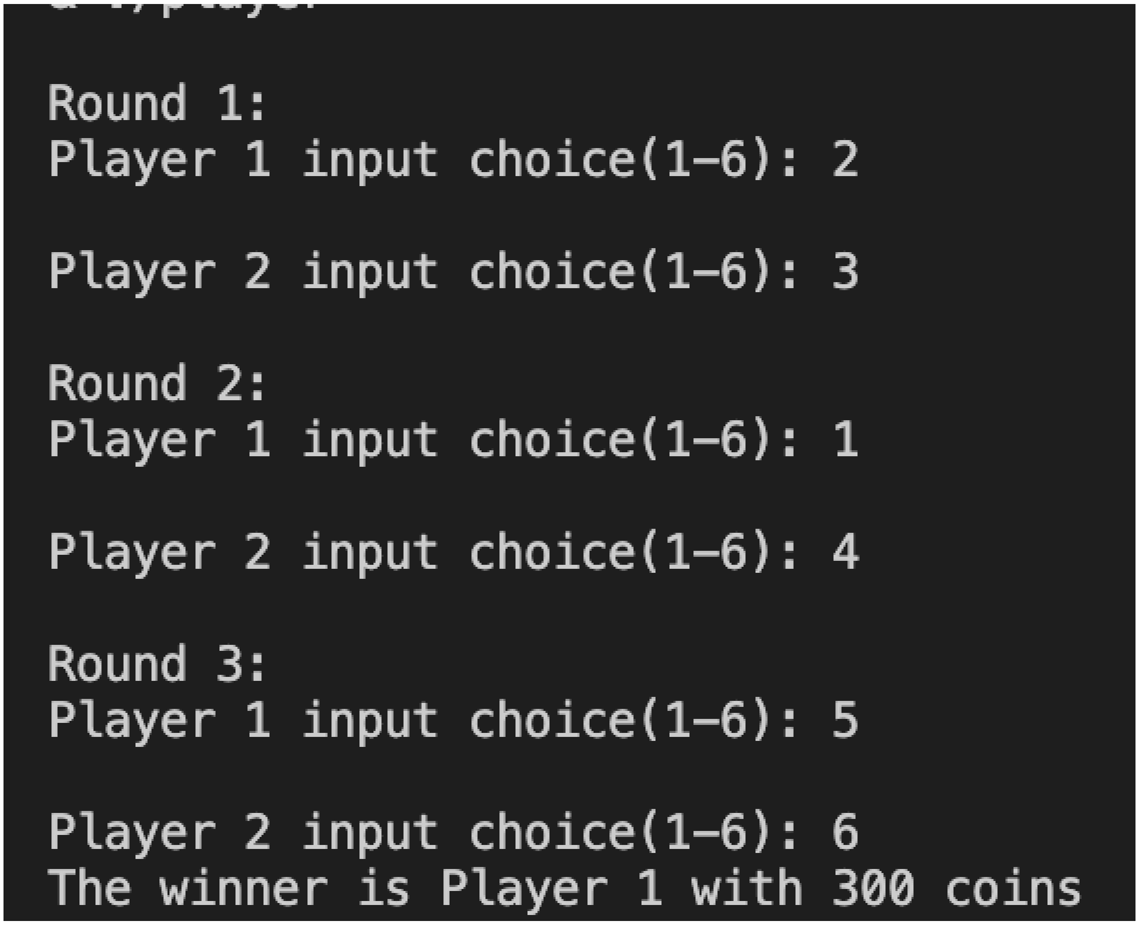
```
#ifndef PLAYER_H
#define PLAYER_H

class Player
{
private:
    int coins;
    int choice;
public:
    Player();
    void playround(int dice);
    int getcoins() const;
    void setcoins(int coin);
    int getchoice() const;
    void setchoice(int nchoice);
    void winner(Player p2);
};
int rollDice();

#endif
```

Player.cpp



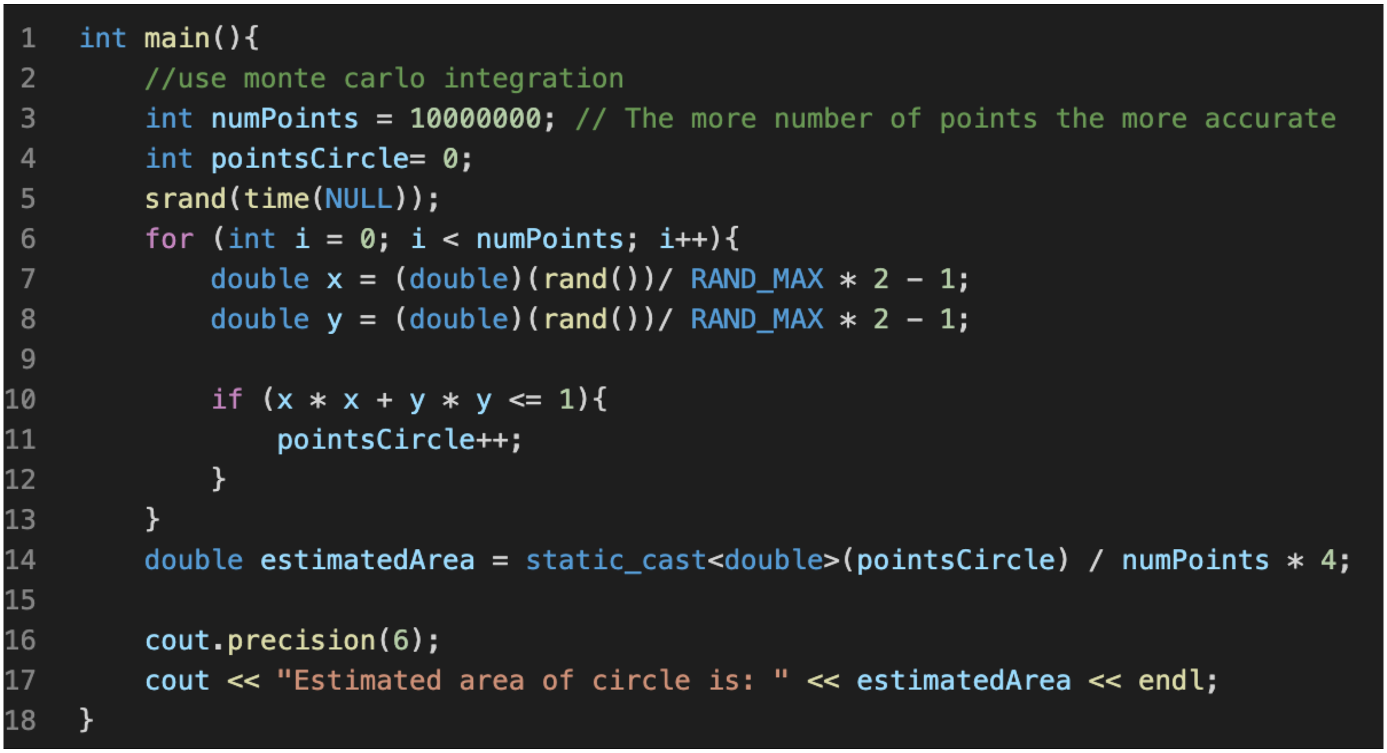
```cpp
Player::Player(){
    int choice = 0;
    int coins = 0;
}

void Player::playround(int dice){
    if (choice == dice)
    {
        coins = coins * 2;
    }
    else
    {
        coins = coins - 50;
    }
}
int Player::getcoins() const{
    return coins;
}
void Player::setcoins(int coin){
    coins = coin;
}
int Player::getchoice() const{
    return choice;
}
void Player::setchoice(int nchoice){
    choice = nchoice;
}
void Player::winner(Player p2){
    if(coins < p2.coins){
        cout << "The winner is Player 1 with " << p2.coins << " coins" <<endl;
    }
    else if(coins > p2.coins){
        cout << "The winner is Player 2 with " << coins << " coins" << endl;
    }
    else{
        cout << "There is a tie with both players having " << coins << " coins" <<endl;
    }
}
int rollDice(){
    return (rand() % 6) + 1;
}
```

Output



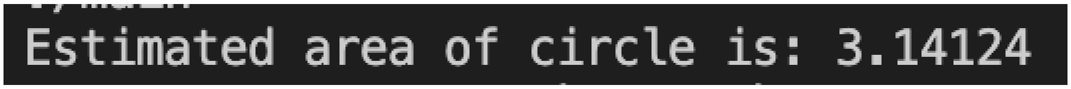
```
Round 1:
Player 1 input choice(1-6): 2

Player 2 input choice(1-6): 3

Round 2:
Player 1 input choice(1-6): 1

Player 2 input choice(1-6): 4

Round 3:
Player 1 input choice(1-6): 5

Player 2 input choice(1-6): 6
The winner is Player 1 with 300 coins
```

Part 5:



```
int main(){
    //use monte carlo integration
    int numPoints = 10000000; // The more number of points the more accurate
    int pointsCircle= 0;
    srand(time(NULL));
    for (int i = 0; i < numPoints; i++){
        double x = (double)(rand())/ RAND_MAX * 2 - 1;
        double y = (double)(rand())/ RAND_MAX * 2 - 1;

        if (x * x + y * y <= 1){
            pointsCircle++;
        }
    }
    double estimatedArea = static_cast<double>(pointsCircle) / numPoints * 4;

    cout.precision(6);
    cout << "Estimated area of circle is: " << estimatedArea << endl;
}
```

Output



```
Estimated area of circle is: 3.14124
```