Lab2 Assignment: Advance Algorithms

Name: Chea Ilong
ID: 100022
Group: 1
Gen: 10

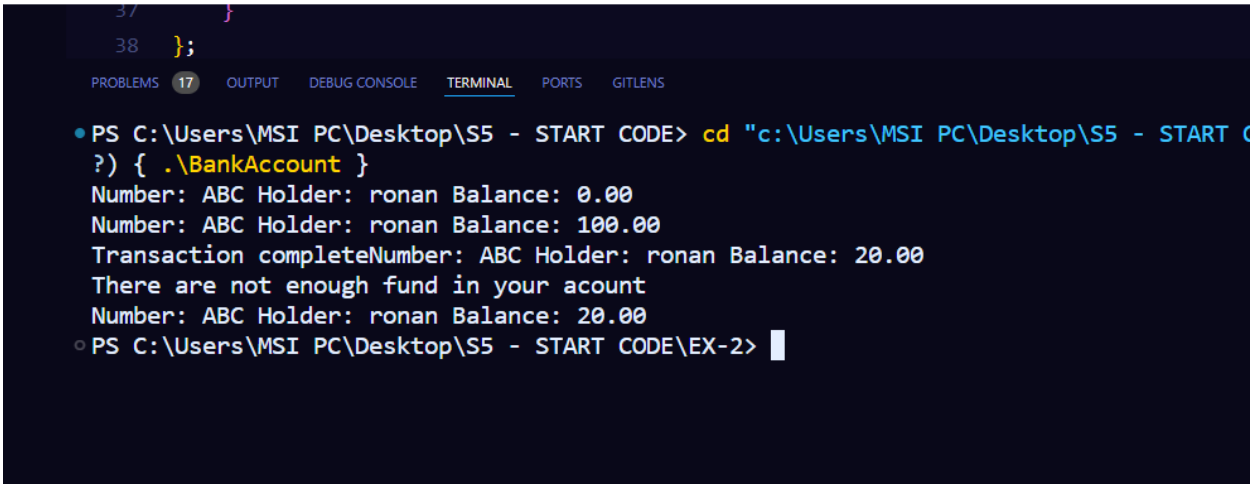
Ex1: Person

```cpp
#include <iostream>
#include <string>
using namespace std;

class Person {
    // Write you code here
    public:
    string lastName;
    string firstName;
    int securityNumber;

    Person (string first, string last, int security){
        this->firstName = first;
        this->lastName = last;
        this->securityNumber = security;
    }
    void setFirstName(string name){
         firstName = name;
    };
    string toString(){
       return  "First Name: "+ firstName + "Last Name:  " + lastName + " Security NUmber: " + to_string(securityNumber);

};

};
int main() {
    Person ronan("ronan", "ogor", 4785);

    ronan.setFirstName("ronano");
    std::cout << ronan.toString() << std::endl;

    return 0;
}

```

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   GITLENS

PS C:\Users\MSI PC\Desktop\S5 - START CODE> cd "c:\Users\MSI PC\Desktop\S5 - START CODE\EX-1\" ; if ($?) { .\Person
First Name: ronano Last Name:  ogor,  Security NUmber: 4785
PS C:\Users\MSI PC\Desktop\S5 - START CODE\EX-1>
```

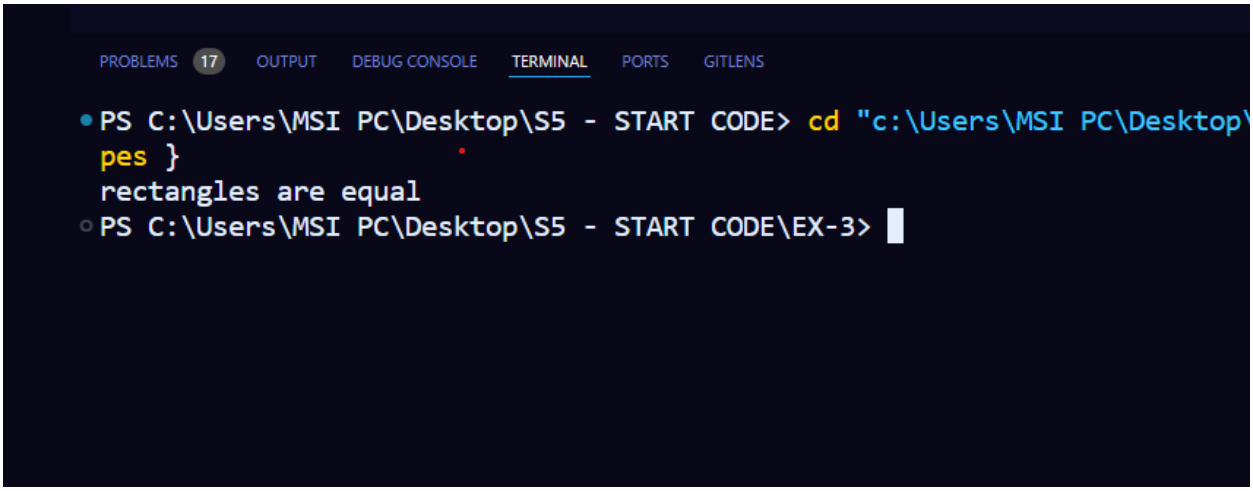
Ex2: Bank

```cpp
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

class BankAccount {
 // TODO
 private:
 string accountNumber;
 string accountholder;
 float balance;

public: BankAccount(string accountNum,string accounthold, float bal){
       this->accountNumber = accountNum;
       this->accountholder = accounthold;
       this->balance = bal;
};
 public: void deposit(float amount){
    balance+= amount;
 };
 public: bool withdraw(float amount){
    if (amount > balance){
        cout << "There are not enough fund in your acount \n";
        return false;
    }else {
        cout << "Transaction complete";
        balance -= amount;
        return true;
    }
 };
    string toString() {
        // Create a string stream to format the output
        stringstream ss;
        ss << fixed << setprecision(2);  // Ensure 2 decimal places
        ss << "Number: " << accountNumber << " Holder: " << accountholder << " Balance: " << balance;
        return ss.str();  // Return the formatted string
    }
};

int main() {

    // Create an account with 0$
    BankAccount myAccount("ABC", "ronan", 0);
    cout<< myAccount.toString() << endl;

    // Deposite 100$
    myAccount.deposit(100);
    cout<< myAccount.toString() << endl;

    // Withdraw 80$ - Should success
    myAccount.withdraw(80);
    cout<< myAccount.toString() << endl;

    // Withdraw 30$ - Should fail
    myAccount.withdraw(30);
    cout<< myAccount.toString() << endl;

    return 0;
}
```


};
PROBLEMS 17
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
PS C:\Users\MSI PC\Desktop\S5 - START CODE> cd "c:\Users\MSI PC\Desktop\S5 - START C
?) { .\BankAccount }
Number: ABC Holder: ronan Balance: 0.00
Number: ABC Holder: ronan Balance: 100.00
Transaction completeNumber: ABC Holder: ronan Balance: 20.00
There are not enough fund in your acount
Number: ABC Holder: ronan Balance: 20.00
PS C:\Users\MSI PC\Desktop\S5 - START CODE\EX-2>

Ex3: Shape

```cpp
#include <iostream>
#include <string>
using namespace std;

class Point2D
{
public:
    double x;
    double y;

    Point2D(double x_val, double y_val) : x(x_val), y(y_val) {}

    bool isEqual(const Point2D &other)
    {
        return this->x == other.x && this->y == other.y;
    }
};

class Rectangle
{
public:
    // TODO
    Point2D bottomLeft;
    double width;
    double height;

    Rectangle(const Point2D &bottomLeft, const Point2D &topright)
        : bottomLeft(bottomLeft)
    {
        this->width = topright.x - bottomLeft.x;
        this->height = topright.y - bottomLeft.y;
    }

    double perimeter()
    {
        return (width + height) * 2;
    }

    double area()
    {
        return width * height;
    }

    bool isEqual(const Rectangle &other)
    {

        return bottomLeft.isEqual(other.bottomLeft) &&
               width == other.width &&
               height == other.height;
    }
};

int main()
{

    Point2D p1(0, 0);
    Point2D p2(10, 20);

    Rectangle r1(p1, p2);
    Rectangle r2(p1, p2);

    string message = r1.isEqual(r2) ? "rectangles are equal" : "rectangles are not equal";
    cout << message << std::endl;

    return 0;
}

```


PROBLEMS 17
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
PS C:\Users\MSI PC\Desktop\S5 - START CODE> cd "c:\Users\MSI PC\Desktop\
pes }
rectangles are equal
PS C:\Users\MSI PC\Desktop\S5 - START CODE\EX-3>