Object Oriented Programming

Assignment # 03

Muhammad Shakaib Arsalan

Student ID: F2022266626

Course Code: CC1022

Section: V2

Resource Person: Rehan Raza

School of Systems and Technology

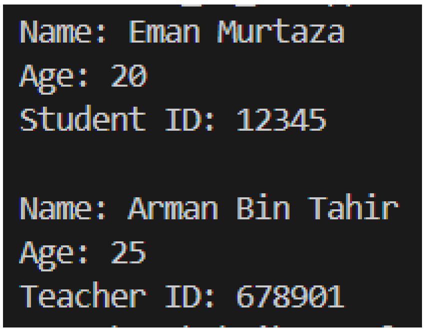
UMT Lahore Pakistan

Problem 01

Code:

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

// Base class Person
class person
{
protected:
    string name;
    int age;

public:
    // Constructor with default values for name and age
    person(string name = "", int age = 0)
    {
        this->name = name;
        this->age = age;
    }

    // Virtual function to display details
    virtual void displayDetails()
    {
        cout << "Name: " << name << "\n";
        cout << "Age: " << age << "\n";
    }
};

// Derived class Student
class student : public person
{
private:
    int studentID;

public:
    // Constructor with default values for name, age, and studentID
    student(string name = "", int age = 0, int studentID = 0) : person(name,
age)
    {
        this->studentID = studentID;
    }

    // Override displayDetails() function to display additional studentID
    void displayDetails()
    {
        cout << "Name: " << name << "\n";
        cout << "Age: " << age << "\n";
        cout << "Student ID: " << studentID << "\n";
    }
};

// Derived class Teacher
class teacher : public person
{
private:
    int teacherID;

public:
    // Constructor with default values for name, age, and teacherID
    teacher(string name = "", int age = 0, int teacherID = 0) : person(name,
age)
    {
        this->teacherID = teacherID;
    }

    // Override displayDetails() function to display additional teacherID
    void displayDetails()
    {
        cout << "Name: " << name << "\n";
        cout << "Age: " << age << "\n";
        cout << "Teacher ID: " << teacherID << "\n";
    }
};

int main()
{
    // Creating objects using base class pointers
    person *per01 = new student("Eman Murtaza", 20, 12345);
    person *per02 = new teacher("Arman Bin Tahir", 25, 678901);

    // Calling displayDetails() function for each object
    per01->displayDetails();
    cout << endl;
    per02->displayDetails();

    // Deleting dynamically allocated objects to free memory
    delete per01;
    delete per02;

    return 0;
}
```

Output:



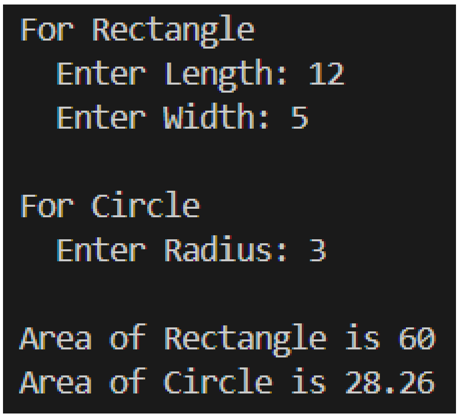
```
Name: Eman Murtaza
Age: 20
Student ID: 12345

Name: Arman Bin Tahir
Age: 25
Teacher ID: 678901
```

Problem 02

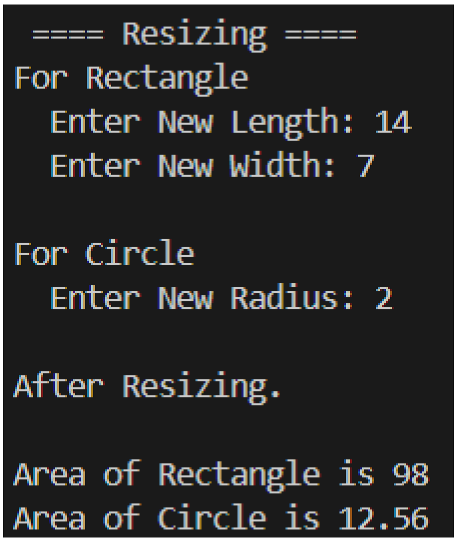
Code:

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

class shape
{
public:
    virtual double calculateArea() =
0;                                         // Pure virtual function for
calculating the area of a shape
    virtual void resize(double width = 0, double height = 0, double radius = 0)
= 0; // Pure virtual function for resizing a shape
};

class rectangle : public shape
{
    double width;  // Private member variable to store the width of the
rectangle
    double height; // Private member variable to store the height of the
rectangle

public:
    rectangle(double width = 0, double height = 0)
    {
        this->width = width;   // Assign the provided width to the member
variable
        this->height = height; // Assign the provided height to the member
variable
    }

    double calculateArea() // Calculate the area of the rectangle
    {
        return width * height;
    }

    void resize(double width = 0, double height = 0, double radius = 0) //
Resize the rectangle by updating its dimensions
    {
        this->width = width;   // Assign the provided width to the member
variable
        this->height = height; // Assign the provided height to the member
variable
    }
};

class circle : public shape
{
    double radius; // Private member variable to store the radius of the circle

public:
    circle(double radius = 0)
    {
        this->radius = radius; // Assign the provided radius to the member
variable
    }

    double calculateArea() // Calculate the area of the circle
    {
        return 3.14 * radius * radius;
    }

    void resize(double radius = 0, double width = 0, double height = 0) //
Resize the circle by updating its radius
    {
        this->radius = radius; // Assign the provided radius to the member
variable
    }
};

int main()
{
    double length = 0, width = 0, radius = 0;
    cout << "For Rectangle\n";
    cout << "  Enter Length: ";
    cin >> length;
    cout << "  Enter Width: ";
    cin >> width;

    cout << "\nFor Circle\n";
    cout << "  Enter Radius: ";
    cin >> radius;
    shape *ptr01 = new rectangle(length, width); // Create a pointer to the
base class shape and instantiate it with a rectangle object
    shape *ptr02 = new circle(radius);              // Create a pointer to the
base class shape and instantiate it with a circle object

    cout << "\nArea of Rectangle is " << ptr01->calculateArea() << endl; //
Call the calculateArea() function on ptr01, which is a rectangle object
    cout << "Area of Circle is " << ptr02->calculateArea() << endl;    // Call
the calculateArea() function on ptr02, which is a circle object

    cout << "\n ==== Resizing ====\n";
    cout << "For Rectangle\n";
    cout << "  Enter New Length: ";
    cin >> length;
    cout << "  Enter New Width: ";
    cin >> width;

    cout << "\nFor Circle\n";
    cout << "  Enter New Radius: ";
    cin >> radius;
    cout << "\nAfter Resizing.\n";
    ptr01->resize(length,width); // Call the resize() function on ptr01, which
is a rectangle object, to resize its dimensions
    ptr02->resize(radius);      // Call the resize() function on ptr02, which
is a circle object, to resize its radius

    cout << "\nArea of Rectangle is " << ptr01->calculateArea() << endl; //
Call the calculateArea() function on ptr01 again after resizing
    cout << "Area of Circle is " << ptr02->calculateArea() << endl;    // Call
the calculateArea() function on ptr02 again after resizing

    return 0;
}
```

Output:



```
For Rectangle
  Enter Length: 12
  Enter Width: 5

For Circle
  Enter Radius: 3

Area of Rectangle is 60
Area of Circle is 28.26
```



```
 ==== Resizing ====
For Rectangle
  Enter New Length: 14
  Enter New Width: 7

For Circle
  Enter New Radius: 2

After Resizing.

Area of Rectangle is 98
Area of Circle is 12.56
```

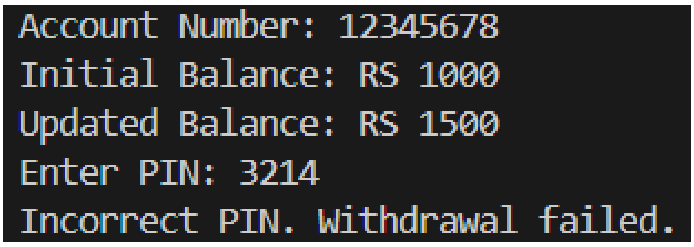
Problem 03

Code:

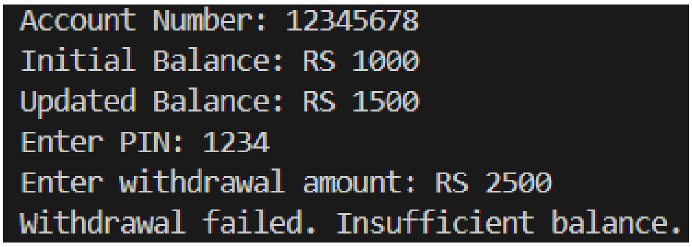
```
#include <iostream>
using namespace std;

// Class representing a Bank Account
class BankAccount
{
private:
    int accountNumber; // Account number
    double balance;    // Account balance
    string pin;        // PIN for account access

public:
    // Constructor to initialize the account number, balance, and PIN
    BankAccount(int accountNumber, double balance, string pin)
    {
        this->accountNumber = accountNumber;
        this->balance = balance;
        this->pin = pin;
    }

    // Getter method for retrieving the account number
    int getAccountNumber()
    {
        return accountNumber;
    }

    // Getter method for retrieving the account balance
    double getBalance()
    {
        return balance;
    }

    // Getter method for retrieving the PIN
    string getPin()
    {
        return pin;
    }

    // Setter method for updating the account balance
    void setBalance(double newBalance)
    {
        balance = newBalance;
    }

    // Method to withdraw funds from the account
    void withdraw(double amount)
    {
        balance -= amount;
        cout << "Withdrawal successful. Remaining balance: RS " << balance <<
endl;
    }
};

int main()
{
    // Create a BankAccount object
    BankAccount myAccount(12345678, 1000.0, "1234");

    // Display account information
    cout << "Account Number: " << myAccount.getAccountNumber() << endl;
    cout << "Initial Balance: RS " << myAccount.getBalance() << endl;

    // Update the account balance
    myAccount.setBalance(1500.0);
    cout << "Updated Balance: RS " << myAccount.getBalance() << endl;

    // Perform a withdrawal from the account
    string pin;
    double withdrawalAmount;

    cout << "Enter PIN: ";
    cin >> pin;

    // Check if the entered PIN is correct
    if (pin == myAccount.getPin())
    {
        cout << "Enter withdrawal amount: RS ";
        cin >> withdrawalAmount;

        // Check if the withdrawal amount is less than or equal to the account
balance
        if (withdrawalAmount <= myAccount.getBalance())
        {
            myAccount.withdraw(withdrawalAmount);
        }
        else
        {
            cout << "Withdrawal failed. Insufficient balance.\n";
        }
    }
    else
    {
        cout << "Incorrect PIN. Withdrawal failed.\n";
    }

    return 0;
```

```
}
```

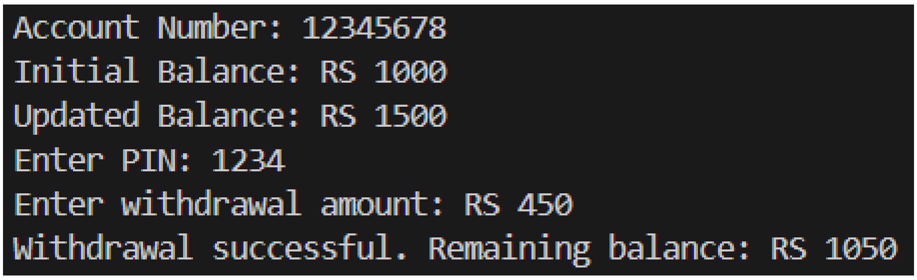
Output:



```
Account Number: 12345678
Initial Balance: RS 1000
Updated Balance: RS 1500
Enter PIN: 3214
Incorrect PIN. Withdrawal failed.
```



```
Account Number: 12345678
Initial Balance: RS 1000
Updated Balance: RS 1500
Enter PIN: 1234
Enter withdrawal amount: RS 2500
Withdrawal failed. Insufficient balance.
```



```
Account Number: 12345678
Initial Balance: RS 1000
Updated Balance: RS 1500
Enter PIN: 1234
Enter withdrawal amount: RS 450
Withdrawal successful. Remaining balance: RS 1050
```

< End >