Lab 2 Codes

Qsn 1:

```cpp
#include <iostream>
#include <string>

using namespace std;

class Person {
private:
    string name;
    int age;

public:
    Person(string n, int a) {
        name = n;
        age = a;
    }
    void setName(string n) {
        name = n;
    }
    void setAge(int a) {
        age = a;
    }
    string getName() {
        return name;
    }
    int getAge() {
        return age;
    }
};

int main() {
    Person person1("Yuvraj", 20);
    Person person2("Shreyas", 19);

    cout << "Person 1 - Name: " << person1.getName() << ", Age: " << person1.getAge() << endl;
    cout << "Person 2 - Name: " << person2.getName() << ", Age: " << person2.getAge() << endl;
    person1.setName("Trijay");
    person1.setAge(69);

    cout << "Updated Person 1 - Name: " << person1.getName() << ", Age: " << person1.getAge() << endl;

    return 0;
}
```

Qsn 2:

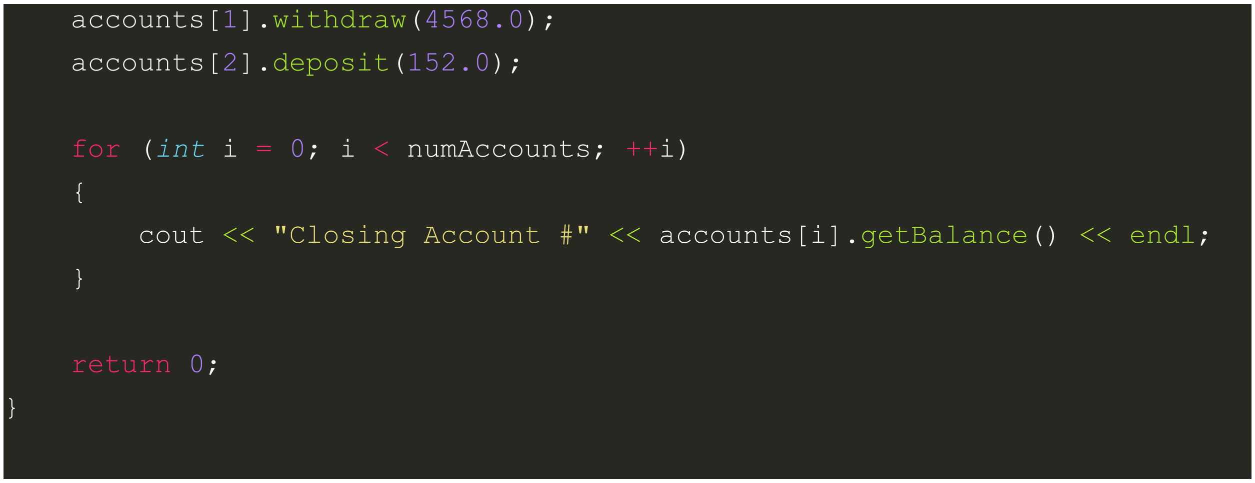
```
#include <iostream>

using namespace std;

class MathUtils
{
public:
    static int add(int a, int b)
    {
        return a + b;
    }

    static int gcd(int a, int b)
    {
        while (b != 0)
        {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
};

int main()
{

    int sum = MathUtils::add(75, 86);
    int greatestCommonDivisor = MathUtils::gcd(34, 85);

    cout << "Sum: " << sum << endl;
    cout << "GCD: " << greatestCommonDivisor << endl;

    return 0;
}
```

Qsn 3:

```
#include <iostream>

using namespace std;

class Rectangle {
private:
    int length;
    int width;

public:

    Rectangle(int len = 1, int wid = 1) {
        length = len;
        width = wid;
    }

    int calcarea() {
        return length * width;
    }

    int calcperi() {
        return 2 * (length + width);
    }
};

int main() {
    Rectangle square;
    cout << "Square Area: " << square.calcarea() << endl;
    cout << "Square Perimeter: " << square.calcperi() << endl;
    Rectangle rectangle(2, 9);
    cout << "Rectangle Area: " << rectangle.calcarea() << endl;
    cout << "Rectangle Perimeter: " << rectangle.calcperi() << endl;

    return 0;
}
```

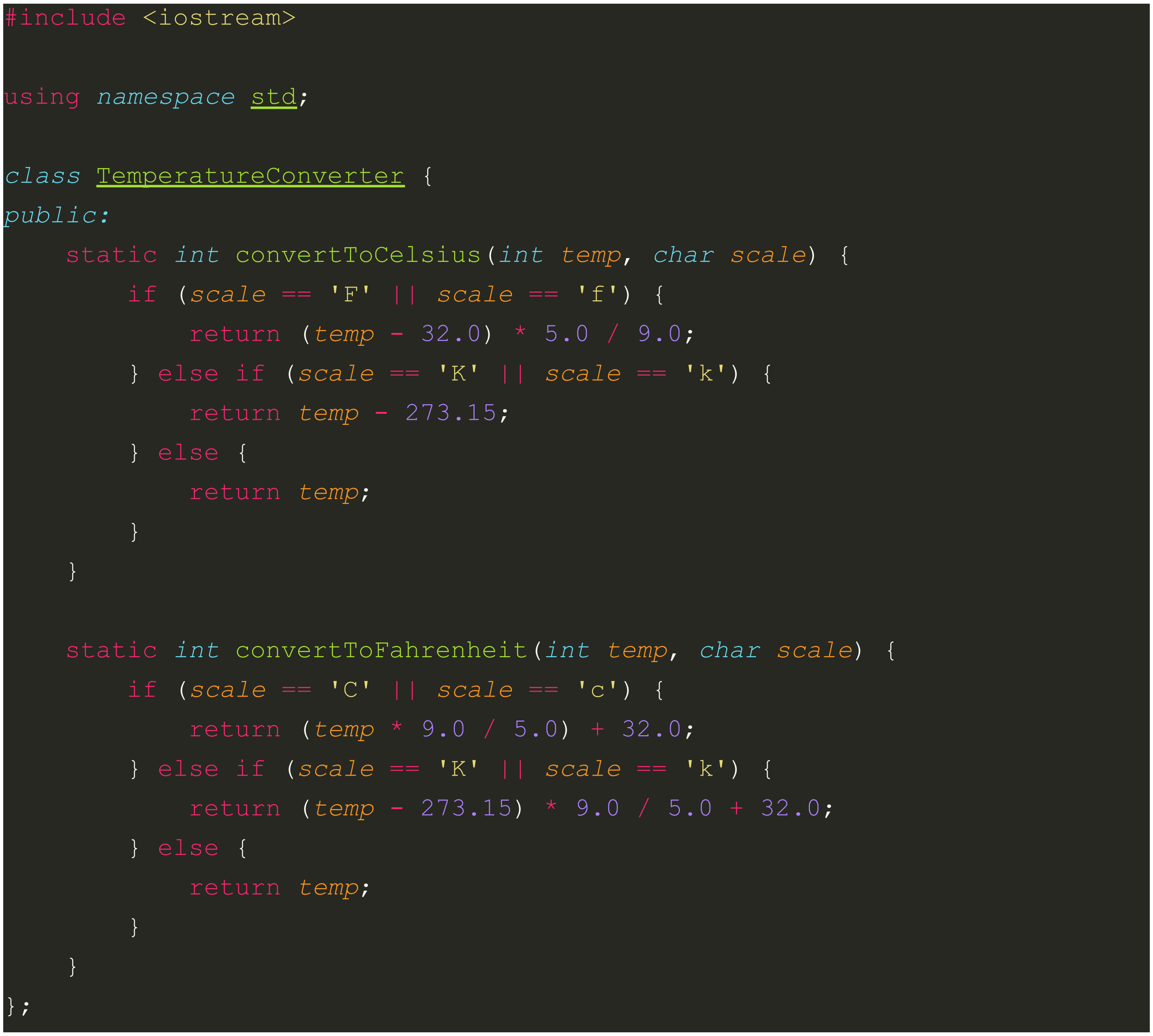
Qsn 4:

```
#include <iostream>

using namespace std;

class BankAccount
{
private:
    int accountNumber;
    int balance;

public:
    BankAccount(int accNumber, double initialBalance)
        : accountNumber(accNumber), balance(initialBalance)
    {
        cout << "Account #" << accountNumber << " opened with initial
balance $" << balance << endl;
    }

    ~BankAccount()
    {
        cout << "Account #" << accountNumber << " is being closed. Final
balance: $" << balance << endl;
    }

    void deposit(double amount)
    {
        if (amount > 0)
        {
            balance += amount;
            cout << "Deposited $" << amount << " into Account #" <<
accountNumber << endl;
        }
        else
        {
            cout << "Invalid deposit amount for Account #" <<
accountNumber << endl;
        }
    }

    void withdraw(double amount)
    {
        if (amount > 0 && balance >= amount)
        {
            balance -= amount;
            cout << "Withdrawn $" << amount << " from Account #" <<
accountNumber << endl;
        }
        else
        {
            cout << "Invalid withdrawal or insufficient balance for
Account #" << accountNumber << endl;
        }
    }

    double getBalance() const
    {
        return balance;
    }
};

int main()
{
    const int numAccounts = 3;
    BankAccount accounts[numAccounts] = {
        BankAccount(69, 548.0),
        BankAccount(420, 1792.0),
        BankAccount(98, 32651.0)};

    accounts[0].deposit(586.0);
```



```
    accounts[1].withdraw(4568.0);
    accounts[2].deposit(152.0);

    for (int i = 0; i < numAccounts; ++i)
    {
        cout << "Closing Account #" << accounts[i].getBalance() << endl;
    }

    return 0;
}
```

Qsn 5:



```
#include <iostream>

using namespace std;

class TemperatureConverter {
public:
    static int convertToCelsius(int temp, char scale) {
        if (scale == 'F' || scale == 'f') {
            return (temp - 32.0) * 5.0 / 9.0;
        } else if (scale == 'K' || scale == 'k') {
            return temp - 273.15;
        } else {
            return temp;
        }
    }

    static int convertToFahrenheit(int temp, char scale) {
        if (scale == 'C' || scale == 'c') {
            return (temp * 9.0 / 5.0) + 32.0;
        } else if (scale == 'K' || scale == 'k') {
            return (temp - 273.15) * 9.0 / 5.0 + 32.0;
        } else {
            return temp;
        }
    }
};
```

```
int main() {
    char iniScale, finScale;
    int temp;

    cout << "Enter temp: ";
    cin >> temp;
    cout << "Enter the source temp scale (C/F/K): ";
    cin >> iniScale;
    cout << "Enter the target temp scale (C/F/K): ";
    cin >> finScale;

    int convTemp;

    if (iniScale == finScale) {
        convTemp = temp;
    } else if ((iniScale == 'C' || iniScale == 'c') && (finScale == 'F' ||
finScale == 'f')) {
        convTemp = TemperatureConverter::convertToFahrenheit(temp,
iniScale);
    } else if ((iniScale == 'F' || iniScale == 'f') && (finScale == 'C' ||
finScale == 'c')) {
        convTemp = TemperatureConverter::convertToCelsius(temp, iniScale);
    } else {
        cout << "Invalid conversion." << endl;
        return 1;
    }

    cout << "Converted temp: " << convTemp << " " << finScale << endl;

    return 0;
}
```

Qsn 6:

```
#include <iostream>
#include <string>

using namespace std;
```

```
class Employee {
private:
    string name;
    int ID;
    string department;


public:

    Employee(std::string name = "Unknown", int ID = 0, std::string
department = "Unknown")
        : name(name), ID(ID), department(department) {

    }


    void displayInfo() {
        cout << "Name: " << name << endl;
        cout << "ID: " << ID << endl;
        cout << "Department: " << department << endl;

    }
};

int main() {

    Employee emp1;
    Employee emp2("Shreyas Ladhe", 11081, "HR");

    cout << "Employee 1 Information:" << endl;
    emp1.displayInfo();
    cout << endl;

    cout << "Employee 2 Information:" << endl;
    emp2.displayInfo();

    return 0;
}
```