**National University of Technology**



**Data Structure and Algorithms**

**Instructor Name: Ms Tabinda Nasir**   **Department:** Artificial Intelligence

**Batch: AI** 2022  **Assessment Type:** Lab Task

**Session:** Fall 2023  **Assignment Number:** 01

**Name:** Syed M. Zeeshan   **Reg No:** F22607017

**Task 1:**

Write a program that allows the user an amount and floating-point number

representing tax percent, and then displays the sale tax.

(Hint: Sale Tax = amount × (tax percent / 100))

**Code:**

#include <iostream>

using namespace std;

int main() {

double amount, taxPercent, saleTax;

cout << "Enter the amount: ";

cin >> amount;

cout << "Enter the tax percent: ";

cin >> taxPercent;

saleTax = amount \* (taxPercent / 100);

cout << "Sale Tax: " << saleTax << endl;

return 0;

}

**Output:**

A screenshot of a computer

Description automatically generated

**Task 2:**

Write a program that allows the user to enter a no. of years, and then

displays the corresponding no. of minutes and no. of seconds.

(Hint: minutes = year x 365 x 24 x 60 &amp; seconds = year x 365 x 24 x 60 x 60)

CODE:

#include <iostream>

using namespace std;

int main() {

int years;

long long minutes, seconds;

cout << "Enter the number of years: ";

cin >> years;

minutes = years \* 365 \* 24 \* 60;

seconds = years \* 365 \* 24 \* 60 \* 60;

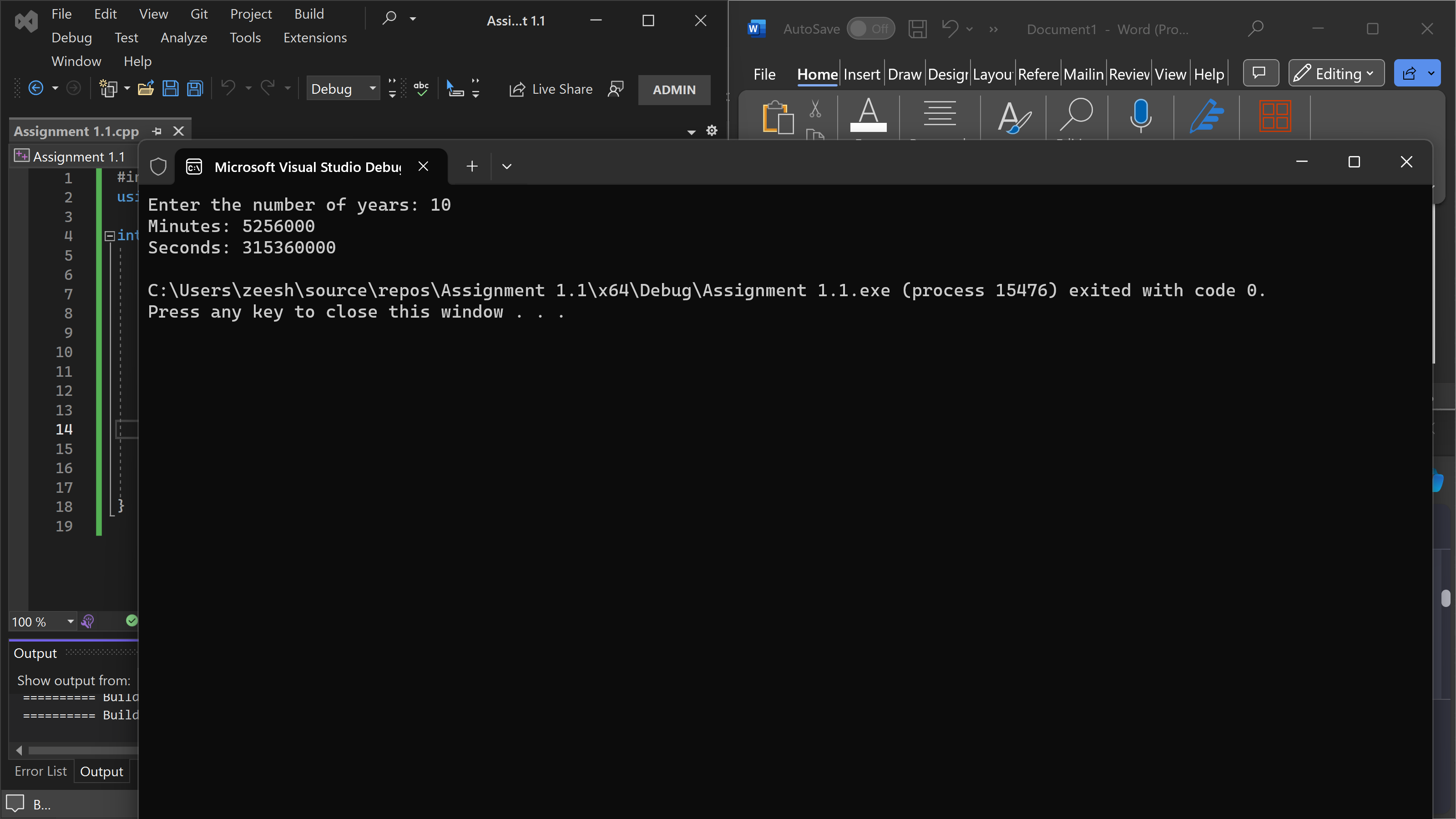
cout << "Minutes: " << minutes << endl;

cout << "Seconds: " << seconds << endl;

return 0;

}

OUTPUT:



TASK 3:

Write a program to transpose any matrix, you have to first ask to the user to enter the matrix and replace row by column and column by row to transpose that matrix, then display the transpose of the matrix on the screen.

CODE:

#include <iostream>

int main() {

int rows, cols;

std::cout << "Enter the number of rows: ";

std::cin >> rows;

std::cout << "Enter the number of columns: ";

std::cin >> cols;

// Dynamic memory allocation for the matrix

int\*\* matrix = new int\* [rows];

for (int i = 0; i < rows; ++i) {

matrix[i] = new int[cols];

}

std::cout << "Enter the matrix elements:" << std::endl;

for (int i = 0; i < rows; ++i) {

for (int j = 0; j < cols; ++j) {

std::cin >> matrix[i][j];

}

}

std::cout << "Original Matrix:" << std::endl;

for (int i = 0; i < rows; ++i) {

for (int j = 0; j < cols; ++j) {

std::cout << matrix[i][j] << " ";

}

std::cout << std::endl;

}

std::cout << "Transpose Matrix:" << std::endl;

for (int i = 0; i < cols; ++i) {

for (int j = 0; j < rows; ++j) {

std::cout << matrix[j][i] << " ";

}

std::cout << std::endl;

}

// Deallocate the dynamically allocated memory

for (int i = 0; i < rows; ++i) {

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

OUTPUT:

A screenshot of a computer

Description automatically generated

TASK 4:

Write a program in C++ to demonstrate an information system for the employees working in AUIC. Employee class contains subclasses for Manager and Teacher. Teacher class has further subclasses for RegularyTeacher and VisitingTeacher. The attributes common by each employee include the name and job title of the employee and date of joining. The regular teachers need an attribute for monthly salary and the visiting teachers have pay rate and hours worked in a month. Both types of teacher have an attribute for qualification. Manager class has attribute for total experience.

CODE:

#include <iostream>

#include <string>

using namespace std;

class Employee {

protected:

string name;

string jobTitle;

string dateOfJoining;

public:

void getData() {

cout << "Enter name: ";

getline(cin, name);

cout << "Enter job title: ";

getline(cin, jobTitle);

cout << "Enter date of joining: ";

getline(cin, dateOfJoining);

}

void displayData() {

cout << "Name: " << name << endl;

cout << "Job Title: " << jobTitle << endl;

cout << "Date of Joining: " << dateOfJoining << endl;

}

};

class Teacher : public Employee {

protected:

string qualification;

public:

void getQualification() {

cout << "Enter qualification: ";

getline(cin, qualification);

}

void displayQualification() {

cout << "Qualification: " << qualification << endl;

}

};

class RegularTeacher : public Teacher {

protected:

double monthlySalary;

public:

void getMonthlySalary() {

cout << "Enter monthly salary: ";

cin >> monthlySalary;

}

void displayMonthlySalary() {

cout << "Monthly Salary: " << monthlySalary << endl;

}

};

class VisitingTeacher : public Teacher {

protected:

double payRate;

int hoursWorked;

public:

void getPayRateAndHours() {

cout << "Enter pay rate: ";

cin >> payRate;

cout << "Enter hours worked in a month: ";

cin >> hoursWorked;

}

void displayPayRateAndHours() {

cout << "Pay Rate: " << payRate << endl;

cout << "Hours Worked: " << hoursWorked << endl;

}

};

class Manager : public Employee {

protected:

int totalExperience;

public:

void getTotalExperience() {

cout << "Enter total experience (in years): ";

cin >> totalExperience;

}

void displayTotalExperience() {

cout << "Total Experience: " << totalExperience << " years" << endl;

}

};

int main() {

RegularTeacher regularTeacher;

VisitingTeacher visitingTeacher;

Manager manager;

cout << "Enter details for Regular Teacher:" << endl;

regularTeacher.getData();

regularTeacher.getQualification();

regularTeacher.getMonthlySalary();

cout << "\nEnter details for Visiting Teacher:" << endl;

visitingTeacher.getData();

visitingTeacher.getQualification();

visitingTeacher.getPayRateAndHours();

cout << "\nEnter details for Manager:" << endl;

manager.getData();

manager.getTotalExperience();

cout << "\nDetails of Regular Teacher:" << endl;

regularTeacher.displayData();

regularTeacher.displayQualification();

regularTeacher.displayMonthlySalary();

cout << "\nDetails of Visiting Teacher:" << endl;

visitingTeacher.displayData();

visitingTeacher.displayQualification();

visitingTeacher.displayPayRateAndHours();

cout << "\nDetails of Manager:" << endl;

manager.displayData();

manager.displayTotalExperience();

return 0;

}

OUTPUT:

