Text

Description automatically generated

**SESD3241-Software Construction and Development**

**Fall 2023 Final Term Lab**

**Time Allowed: 1.5 hours**

|  |  |
| --- | --- |
| Name: |  |
| Registration No |  |

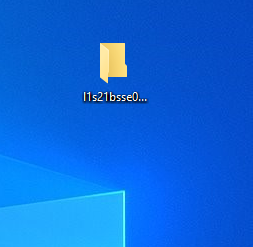
|  |  |  |  |
| --- | --- | --- | --- |
|  | Q1 | Q2 | Overall |
| Total Marks | 20 | 30 | 50 |
| Ob. Marks |  |  |  |

**Instructions:**

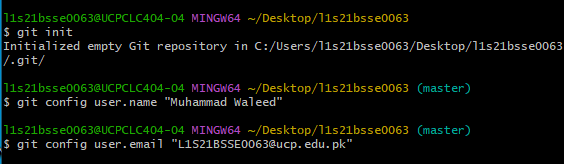
1. **Understanding questions is part of paper**. Therefore, no queries will be entertained during examination.
2. Use proper indentation/formatting while coding.
3. Rename your zip file with your Registration No e.g., **L1F13BSCS0128.zip**
4. **[CLO1] Please read the given scenario carefully then answer the questions given below the scenario [Marks: 20]**

The E-Learning System is designed to provide a seamless online education experience, enabling students to access courses, interact with instructors, and collaborate with peers remotely. Initially it must have a feature of user Authentication where only authorized Students and instructors will be able to log in securely using unique credentials. There will be a section of Course Management where Instructors will be able to create, update, and organize courses with content such as lectures, assignments, and quizzes. The section of Student Enrollment will be making the students able to browse available courses, enroll in desired classes, and gain access to course materials. The important Discussion Forums will be allowing the instructors and students participate in course-specific discussion forums for questions, clarifications, and collaboration. Live Sessions will also be being conducted in Real-time virtual classrooms for live lectures, Q&A sessions, and collaborative activities. The criteria of Assessment and Grading will be allowing the instructors to evaluate students’ performance through quizzes, assignments, and exams, with automated grading features. And the final results, grades and pass-fail status will be shown to the students on their portal.

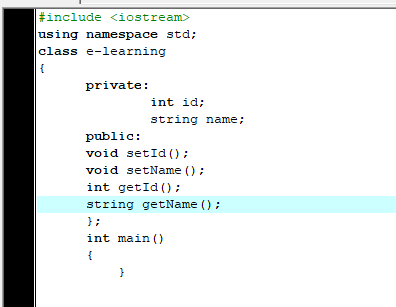
1. Create a project folder on your PC and name it with your roll number. This folder will contain your project files and git repository.



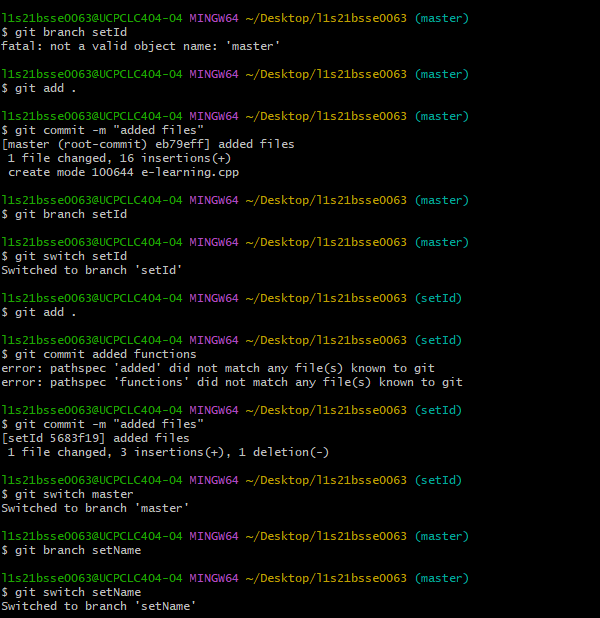
1. Start by initializing a Git repository for your project. Then, configure the commit author specifically for this repository.

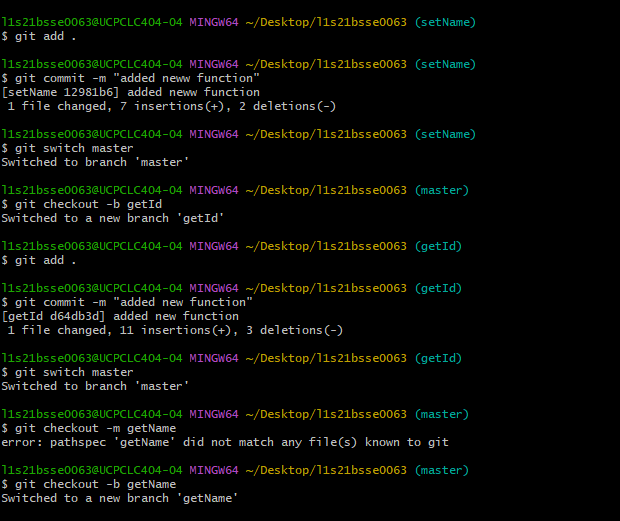


1. Create the main class “E-learning” and perform all the required functions there

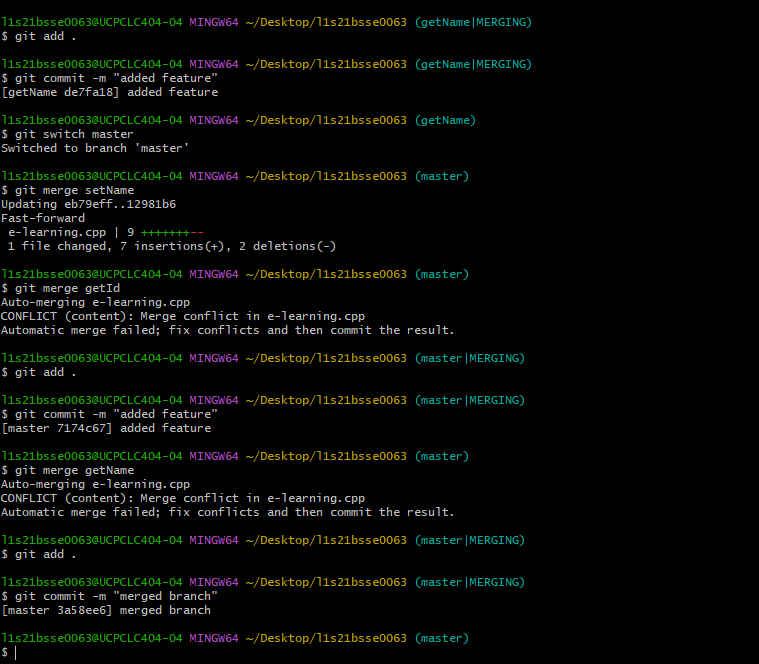


1. Create separate feature branches for each function identified from the above case study, ensuring that each branch is named according to the scenario to reflect the feature it implements.

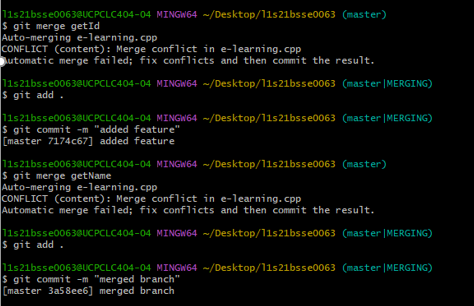




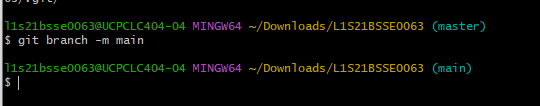
1. Merge each feature branch into the GitHub



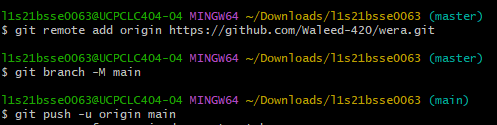
1. Handle any conflicts that may arise during the merging process.



1. You are required to create local and remote repositories. Set the name of the local default branch to “main”.



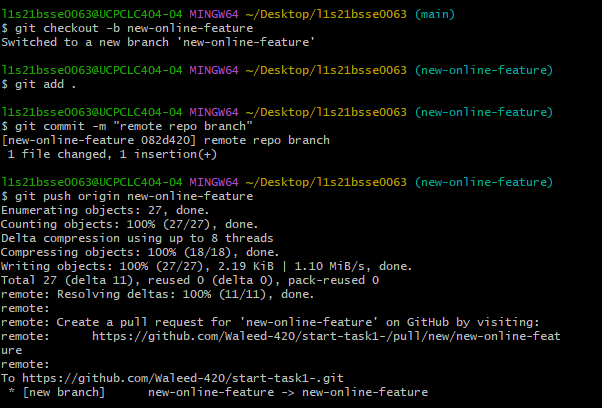
1. Then you are required to establish a connection between remote repository and local repository by setting the remote on the local repository.



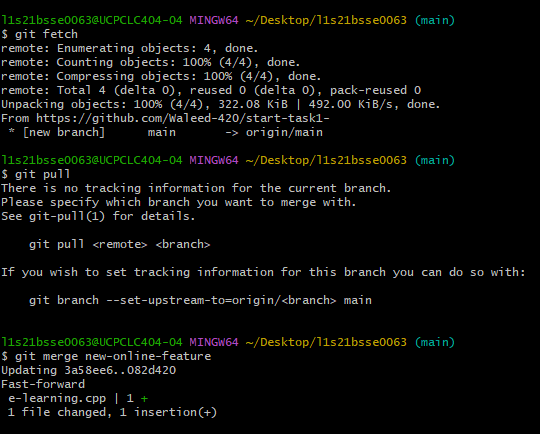
1. Then push the class. Push Your Changes to GitHub. Ensure that your local changes are pushed to your GitHub repository



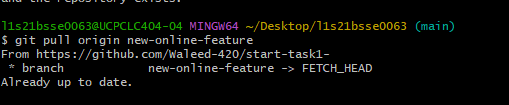
1. A separate feature branch (on their local repository) has to be created for each function. Available in the scenario. Then the feature branch has to be pushed on the remote repository



1. Push your changes to GitHub using the git push command



1. Pull the updated remote repository so that local repository can also be updated accordingly.



**NOTE: After performing each task, take a screen shot and paste it in the word file.**

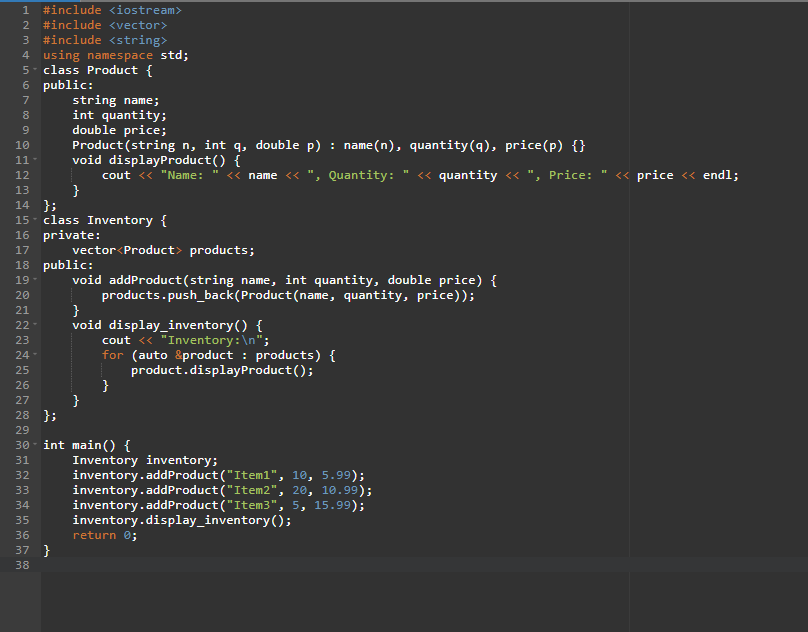
1. **[CLO1]** 
   1. The following Snipped Code has different code smells in it. You are required to specifically name those smells with description. Mention line number on which specific smell is found and briefly justify your answer. [15]

Please Identify the code smells in proper format

**Code Smell-line Number- Justification**

* 1. You are then required to Refactor that code accordingly [15]

**NOTE: Assume all the required header files are included.**



**Code Smells:**

**Long Class:**

The Product class is currently very short, but if it grows in the future, it might be worth breaking it down into smaller, more focused classes. However, as of now, the Product class seems appropriately small.

**Feature Envy:**

Feature Envy occurs when a method of one class is more interested in the internal state of another class than its own. In the current code, the **displayProduct** method in the Product class seems fine, but the **addProduct** and **displayInventory** methods in the Inventory class might be considered as exhibiting feature envy.

**Mysterious Name**

line # 22, 35

The function **display\_inventory** should follow the **camelCase** naming convention.

It's better to rename it to **displayInventory** for consistency.

**Refactored Code:**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

class Product {

public:

string name;

int quantity;

double price;

Product(string n, int q, double p) : name(n), quantity(q), price(p) {}

void displayProduct() const {

cout << "Name: " << name << ", Quantity: " << quantity << ", Price: $" << price << endl;

}

};

class Inventory {

private:

vector<Product> products;

public:

void addProduct(const string& name, int quantity, double price) {

products.push\_back(Product(name, quantity, price));

}

void DisplayInventory() const {

cout << "Inventory:\n";

for (const auto& product : products) {

product.displayProduct();

}

}

};

int main() {

Inventory inventory;

inventory.addProduct("ring", 18, 8.99);

inventory.addProduct("Chain", 28, 2.99);

inventory.addProduct("Robe", 5, 8.99);

inventory.displayInventory();

return 0;

}

**Best Of Luck**