**ISSUES WITH PROCEDURAL PROGRAMMING**



Session: 2022 – 2026

**Submitted by:**

AYESHA KHALID 2022-CS-153

**Supervised by:**

Dr. Awais

Department of Computer Science

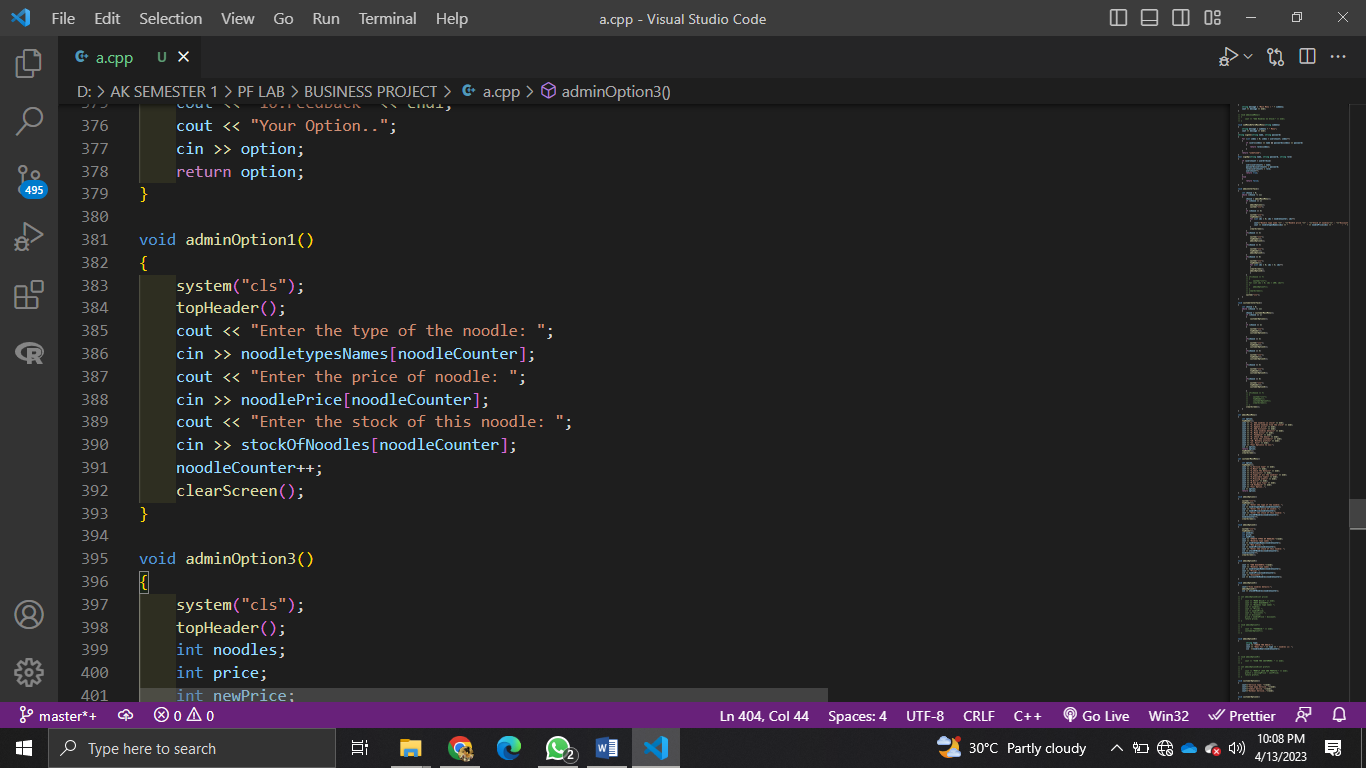
**University of Engineering and Technology**

**Lahore Pakistan**

**Disjoint Data:**

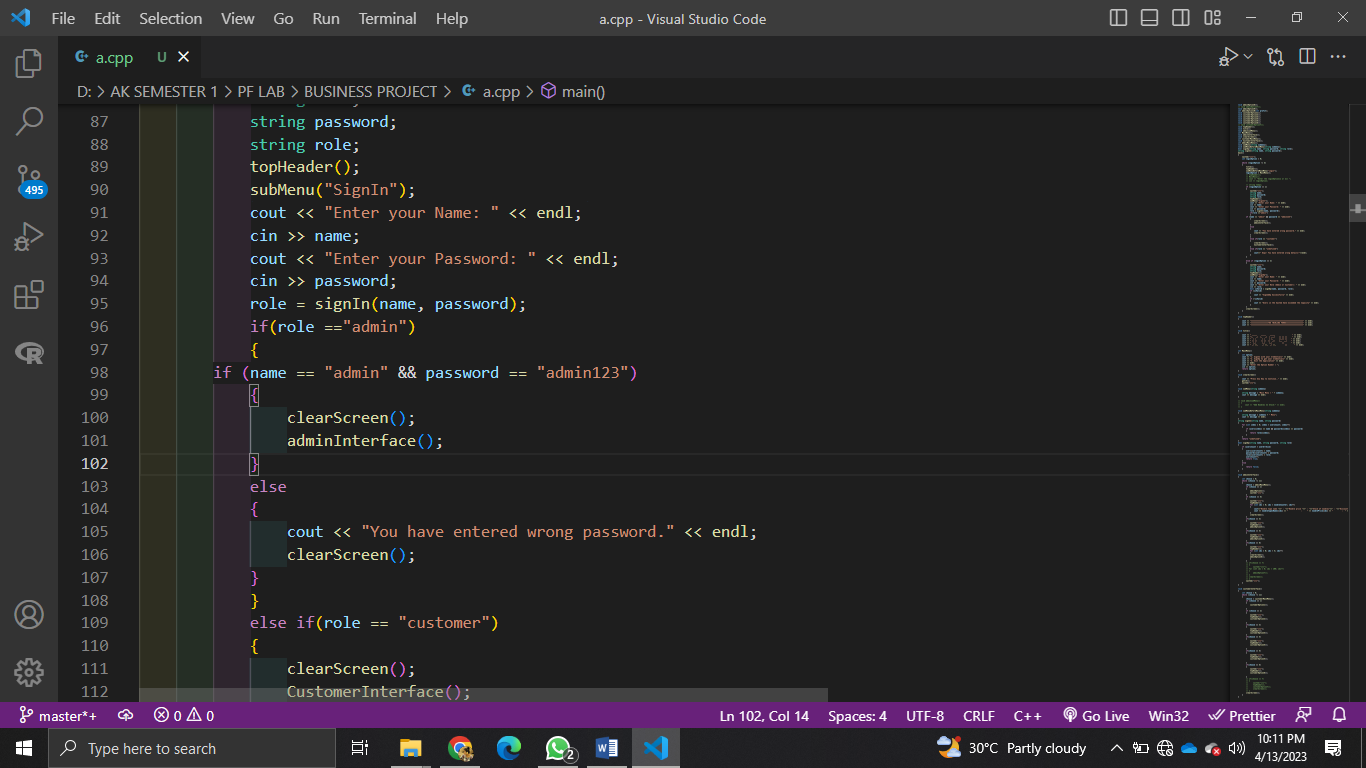
There are many problems in disjoint data. Some of them are discussed below.

**Data Management:**

****                    }

Here I am passing complete array of noodles name, price and stock to a function because each attribute is in different array but in OOP we don’t need to do this . This is a major issue.

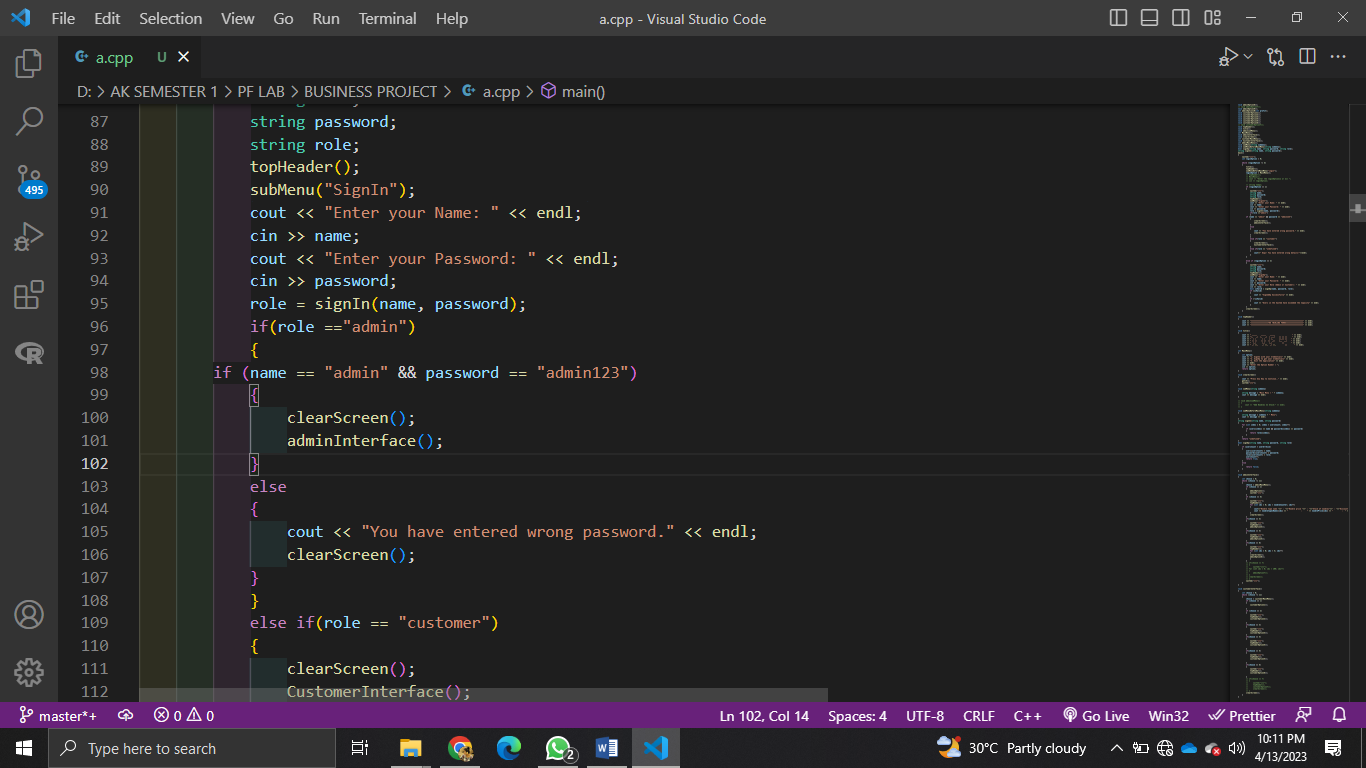
**Constraint on data:**

****<< endl;

In this although name is in string but it cannot restrict the dmin that he should always enter a name not a integer value cannot decide that it should contain a name that is a big security issue using procedural programming and can be resolved by using OOP.

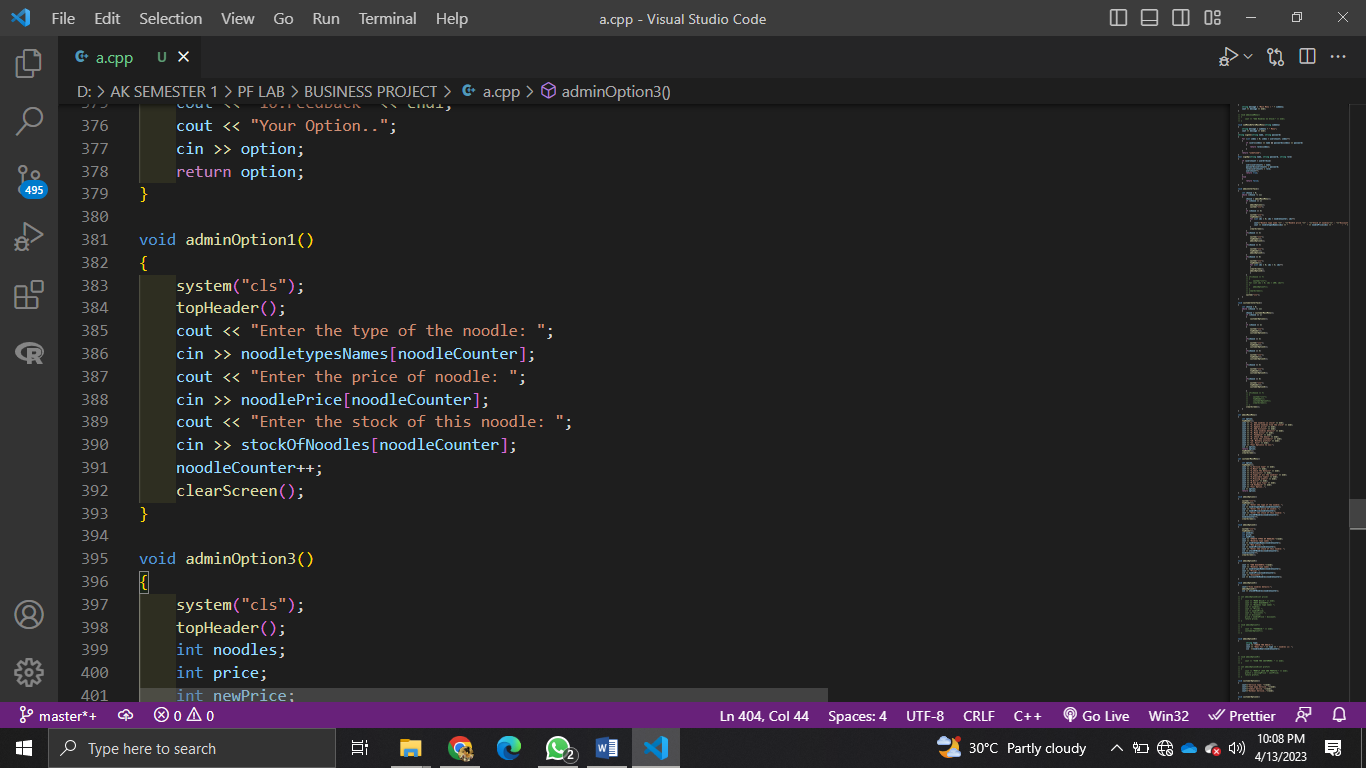
**Complete Access Issues:**

**Example 1:**

****

In this function I am passing complete record of all the users so anyone can access the personal information of any user that is a big security issue using procedural programming and can be resolved by using OOP.

**Extended functionality:**

****

In this function this code only prints the noodles type name, price and stock but we cannot extend its functionality which department that a big issue of using procedural programming and can be resolved by using OOP.

**Game:**

void bulletCollisionWithEnemy1()

{

    for (int x = 0; x < bulletCount; x++)

    {

        if (bulletX[x] + 1 == enemy1X && (bulletY[x] == enemy1Y || bulletY[x] == enemy1Y + 1 || bulletY[x] == enemy1Y + 2 || bulletY[x] == enemy1Y + 3))

        {

            addScore();

            eraseBulletenemy(bulletX[x], bulletY[x]);

            removeBulletFromArrayenemy(x);

            health--;

        }

    }

}

In this function this code only detects bullets collision of Thomas with enemy but not enemy with Thomas. This is lack of extending functionality and that a big issue of using procedural programming and can be resolved by using OOP.

void generateBullet()

{

    char next;

    bulletX[bulletCount] = mainX + 4;

    bulletY[bulletCount] = mainY;

    isBulletActive[bulletCount] = true;

    gotoxy(mainX + 4, mainY);

    cout << "-";

    bulletCount++;

}

In this function this code is only generating the bullet but not moving it but we cannot extend its functionality instead we have to copy the code that a big issue of using procedural programming and can be resolved by using OOP.