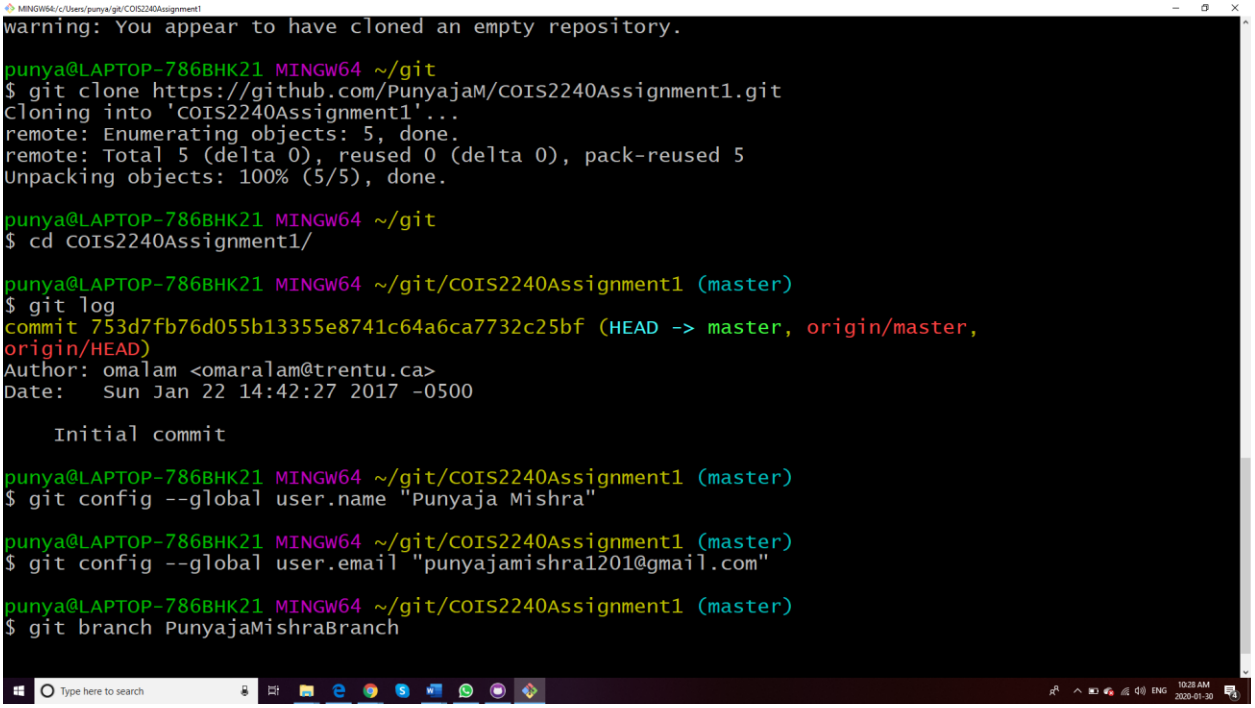
Name : Punyaja Mishra
Student Id : 0660001

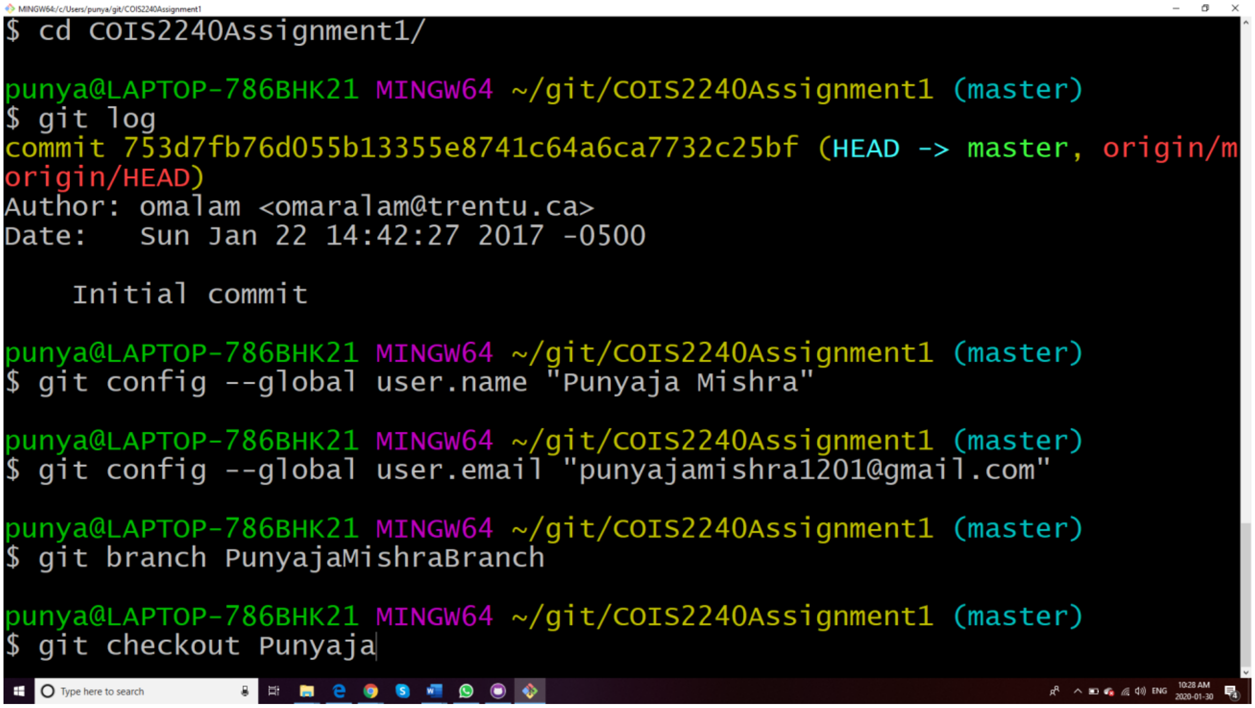
COIS 2020H
Assignment 1

1- **Go to the repository at https://github.com/omalam/COIS2240Assignment1.git and and click Fork. Go to the new repository that GitHub will have created in your account, which will look like https://github.com/<yourname>/COIS2240Assignment1 and click "Clone or Download" and copy URI. In terminal, clone the repository whose URI you just copied**



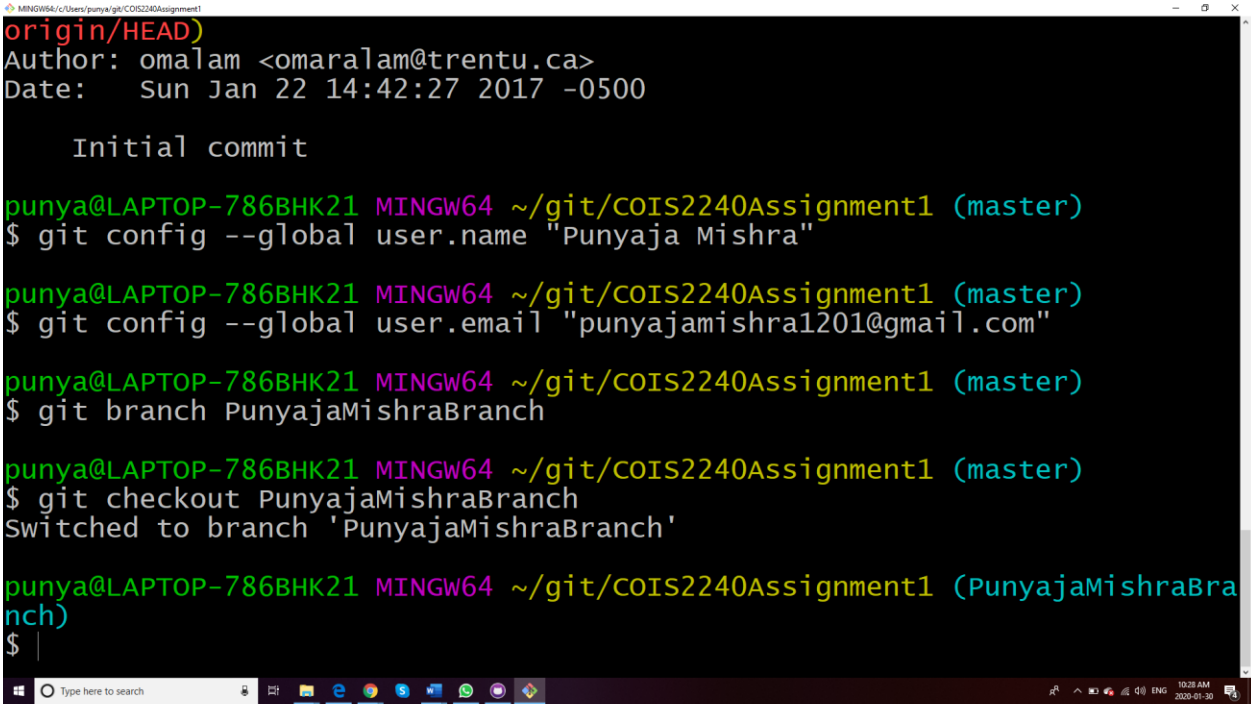
```
warning: You appear to have cloned an empty repository.

punya@LAPTOP-786BHK21 MINGW64 ~/git
$ git clone https://github.com/PunyajaM/COIS2240Assignment1.git
Cloning into 'COIS2240Assignment1'...
remote: Enumerating objects: 5, done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 5
Unpacking objects: 100% (5/5), done.

punya@LAPTOP-786BHK21 MINGW64 ~/git
$ cd COIS2240Assignment1/

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git log
commit 753d7fb76d055b13355e8741c64a6ca7732c25bf (HEAD -> master, origin/master,
origin/HEAD)
Author: omalam <omaralam@trentu.ca>
Date:   Sun Jan 22 14:42:27 2017 -0500

    Initial commit

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git config --global user.name "Punyaja Mishra"

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git config --global user.email "punyajamishra1201@gmail.com"

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git branch PunyajaMishraBranch
```

2- Create a branch using the command line and name it your name. The branch name should be YourFirstNameYourLastName. For example, I can create a branch with my name OmarAlam. You should use the command $ git branch YourFirstNameYourLastNameBranch for this.



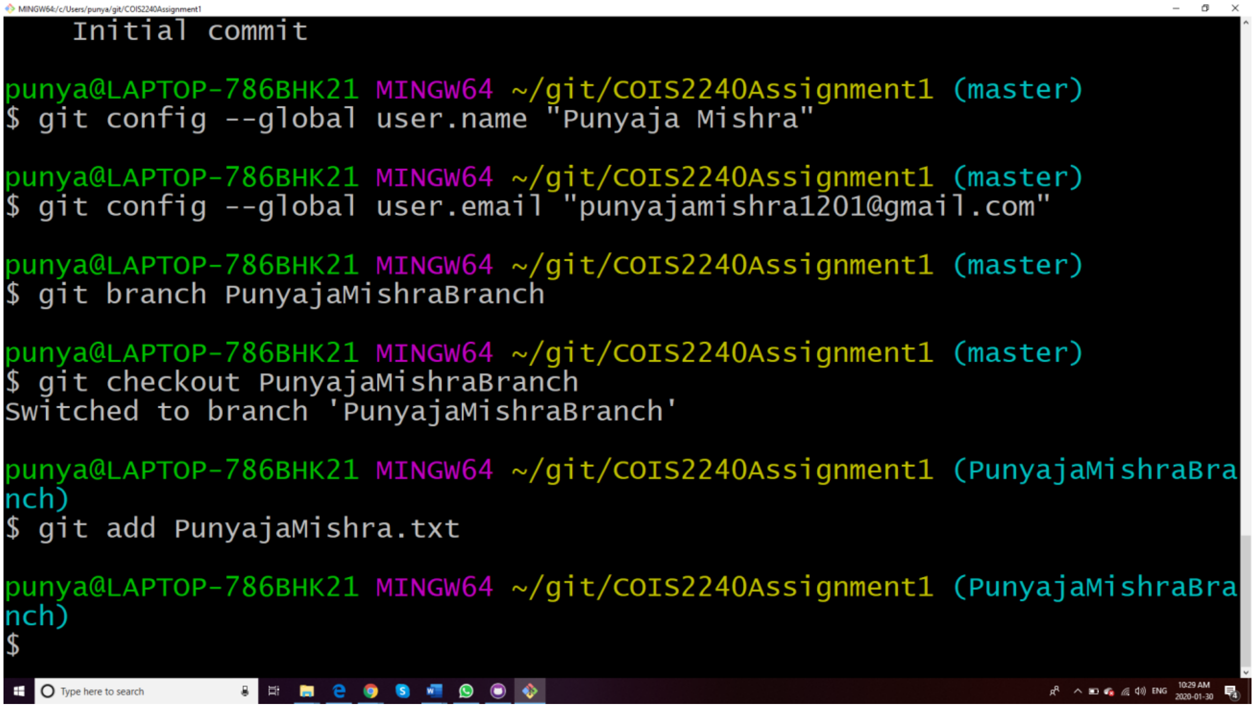
```
$ cd COIS2240Assignment1/

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git log
commit 753d7fb76d055b13355e8741c64a6ca7732c25bf (HEAD -> master, origin/m
origin/HEAD)
Author: omalam <omaralam@trentu.ca>
Date:   Sun Jan 22 14:42:27 2017 -0500

    Initial commit

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git config --global user.name "Punyaja Mishra"

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git config --global user.email "punyajamishra1201@gmail.com"

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git branch PunyajaMishraBranch

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git checkout Punyaja
```

3- Checkout your branch using the command $ git checkout YourFirstNameYourLastNameBranch



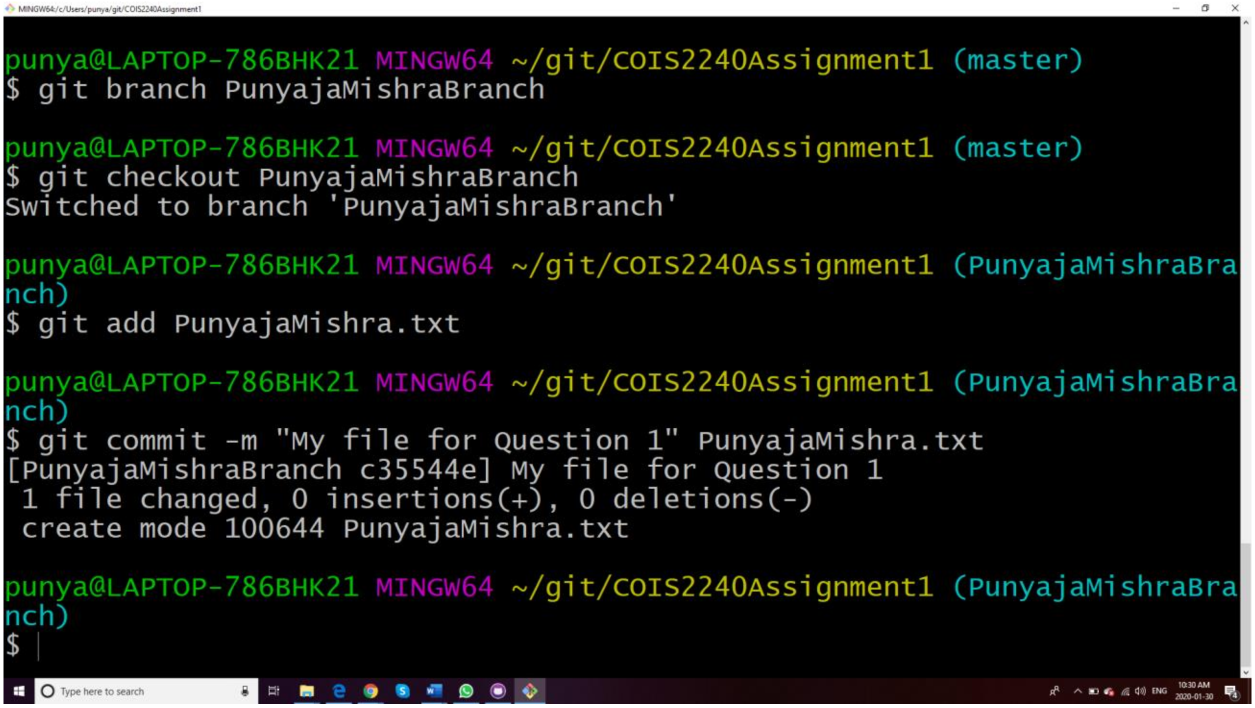
```
origin/HEAD)
Author: omalam <omaralam@trentu.ca>
Date:   Sun Jan 22 14:42:27 2017 -0500

    Initial commit

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git config --global user.name "Punyaja Mishra"

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git config --global user.email "punyajamishra1201@gmail.com"

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git branch PunyajaMishraBranch

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git checkout PunyajaMishraBranch
Switched to branch 'PunyajaMishraBranch'

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (PunyajaMishraBra
nch)
$
```

4- Add an empty text file to your branch using the command $ git add YourFirstNameYourLastName.txt



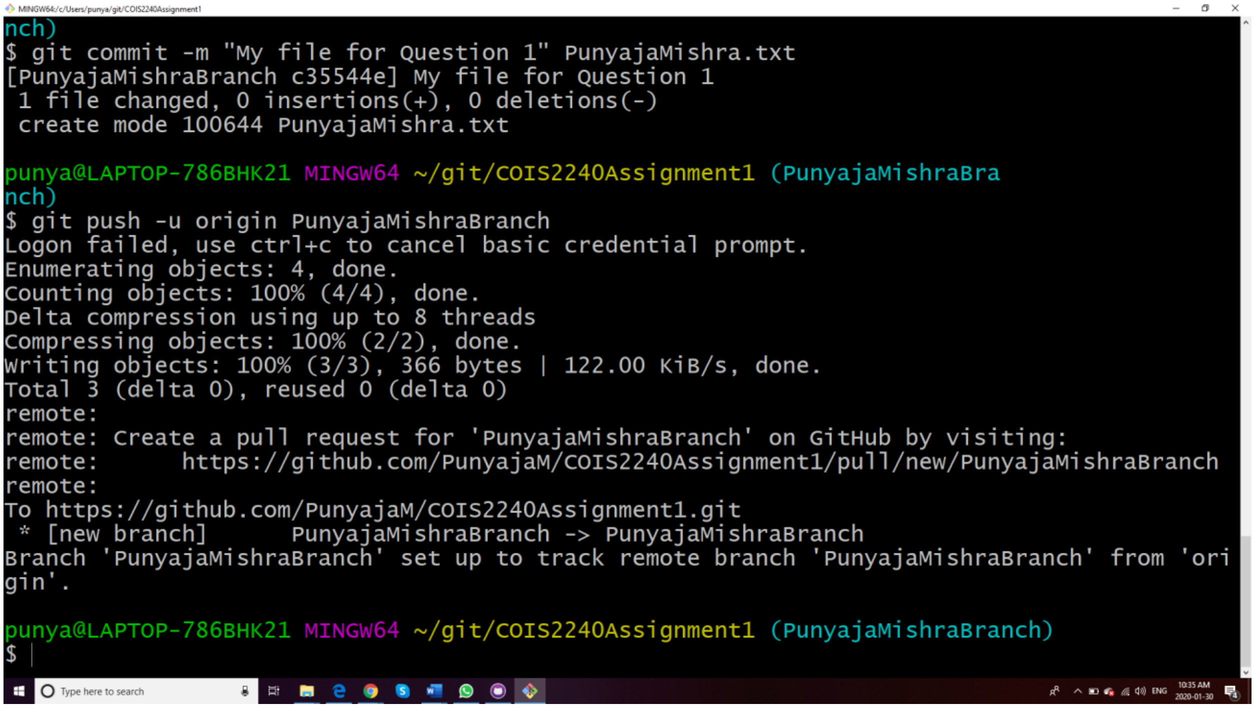
```
    Initial commit

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git config --global user.name "Punyaja Mishra"

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git config --global user.email "punyajamishra1201@gmail.com"

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git branch PunyajaMishraBranch

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git checkout PunyajaMishraBranch
Switched to branch 'PunyajaMishraBranch'

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (PunyajaMishraBra
nch)
$ git add PunyajaMishra.txt

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (PunyajaMishraBra
nch)
$
```

5- Commit the file in the previous using the command $ git commit -m"My file for Question 1" YourFirstNameYourLastName.txt



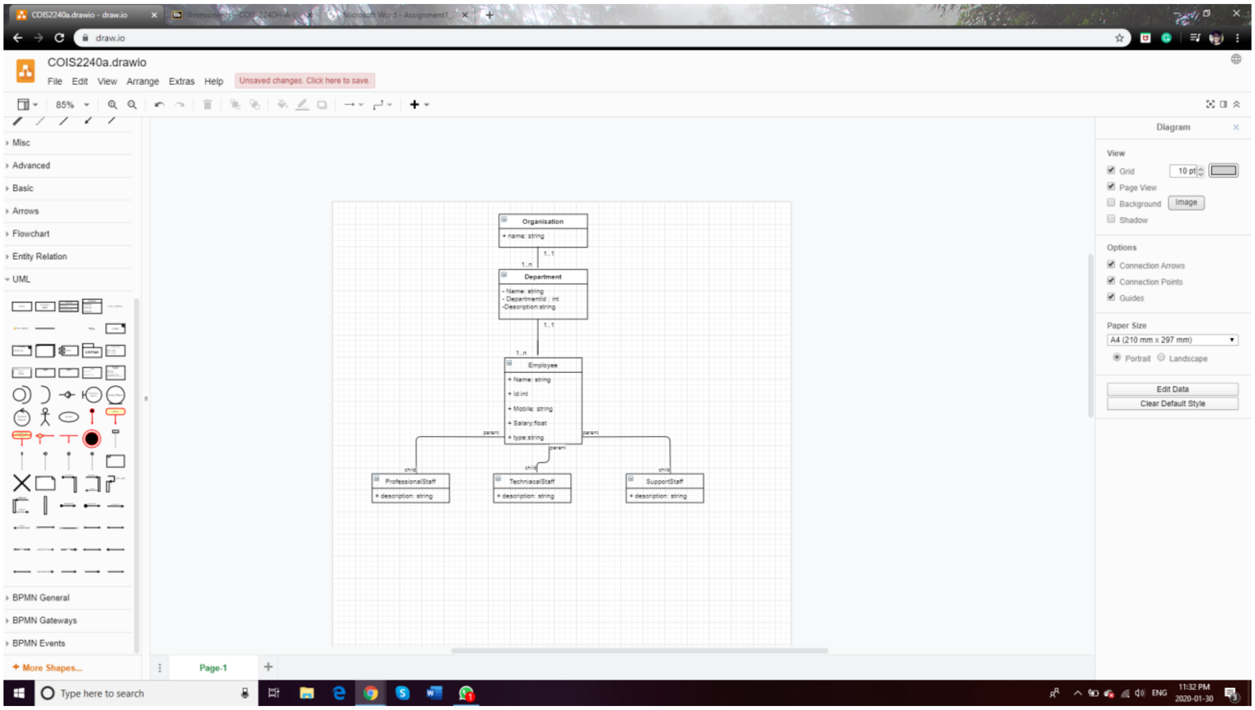
```
punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git branch PunyajaMishraBranch

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (master)
$ git checkout PunyajaMishraBranch
Switched to branch 'PunyajaMishraBranch'

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (PunyajaMishraBra
nch)
$ git add PunyajaMishra.txt

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (PunyajaMishraBra
nch)
$ git commit -m "My file for Question 1" PunyajaMishra.txt
[PunyajaMishraBranch c35544e] My file for Question 1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 PunyajaMishra.txt

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (PunyajaMishraBra
nch)
$
```

6- Push your branch to the remote repository using the command $ git push -u origin YourFirstNameYourLastNameBranch



```
nch)
$ git commit -m "My file for Question 1" PunyajaMishra.txt
[PunyajaMishraBranch c35544e] My file for Question 1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 PunyajaMishra.txt

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (PunyajaMishraBra
nch)
$ git push -u origin PunyajaMishraBranch
Logon failed, use ctrl+c to cancel basic credential prompt.
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 366 bytes | 122.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'PunyajaMishraBranch' on GitHub by visiting:
remote:      https://github.com/PunyajaM/COIS2240Assignment1/pull/new/PunyajaMishraBranch
remote:
To https://github.com/PunyajaM/COIS2240Assignment1.git
 * [new branch]      PunyajaMishraBranch -> PunyajaMishraBranch
Branch 'PunyajaMishraBranch' set up to track remote branch 'PunyajaMishraBranch' from 'ori
gin'.

punya@LAPTOP-786BHK21 MINGW64 ~/git/COIS2240Assignment1 (PunyajaMishraBranch)
$
```

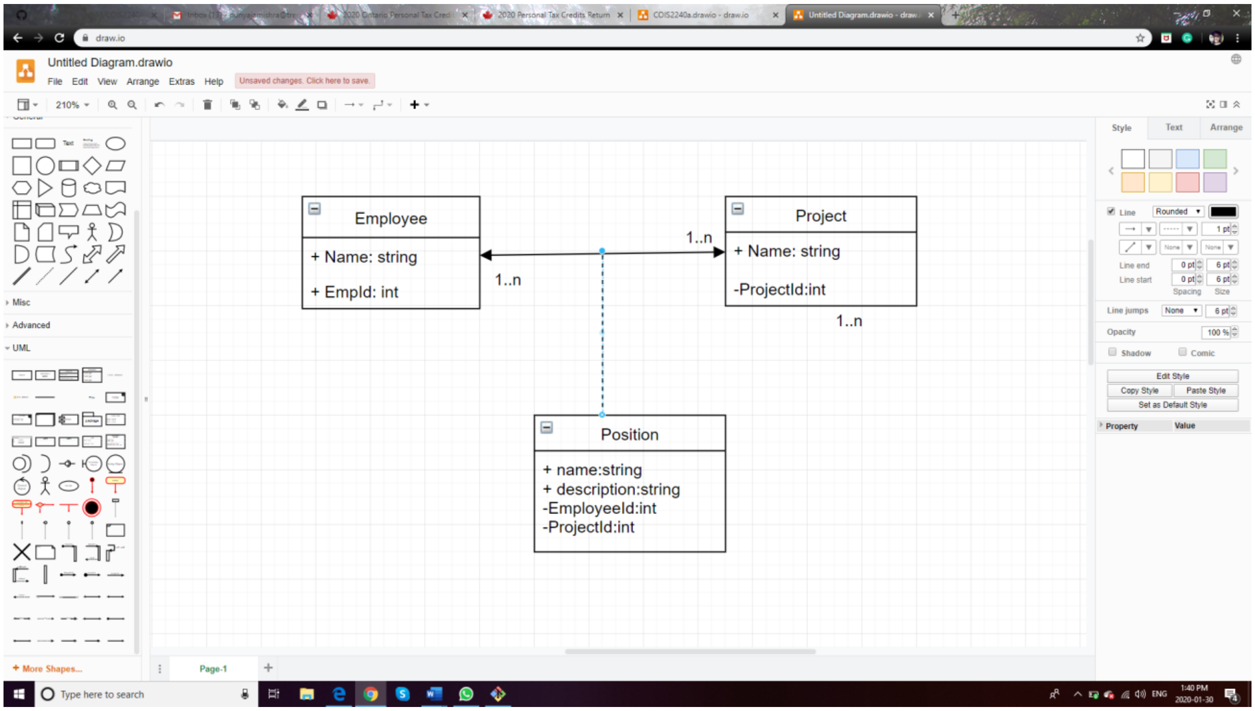
Question 2: (a) An organization has three categories of employee: professional staff, technical staff and support staff. The organization also has departments. Each employee belongs to a department.



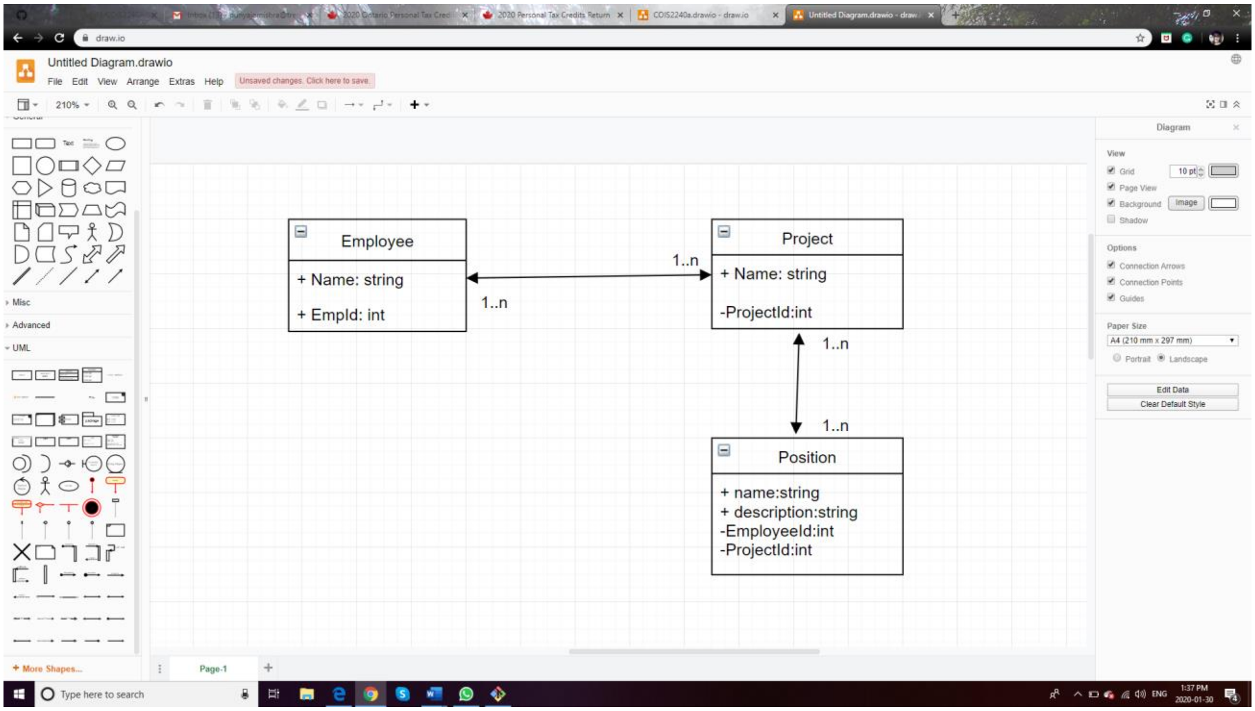
(b) An employee can work on any number of projects and a project can be assigned to any number of employees. However, the employee has a position (role) in the project. The position has a description. Solve this problem with and without using an association class. Employees and Projects should have names.

Draw a class diagrams that captures the above scenarios. You may use any UML drawing tool. There are online UML drawing tools such as draw.io

With association



Without association



Since, every project is going to have a position, in fact many positions, thus they are related. But also, that is the association class because that kinds of reduces the redundancy and the position has an employee and a project associated to it.

Question 3: Problem Description: Design a class named Triangle that extends GeometricObject. The class contains:

• Three double data fields named side1, side2, and side3 with default values 1.0 to denote three sides of the triangle.

• A constructor that creates a triangle with the specified side1, side2, and side3.

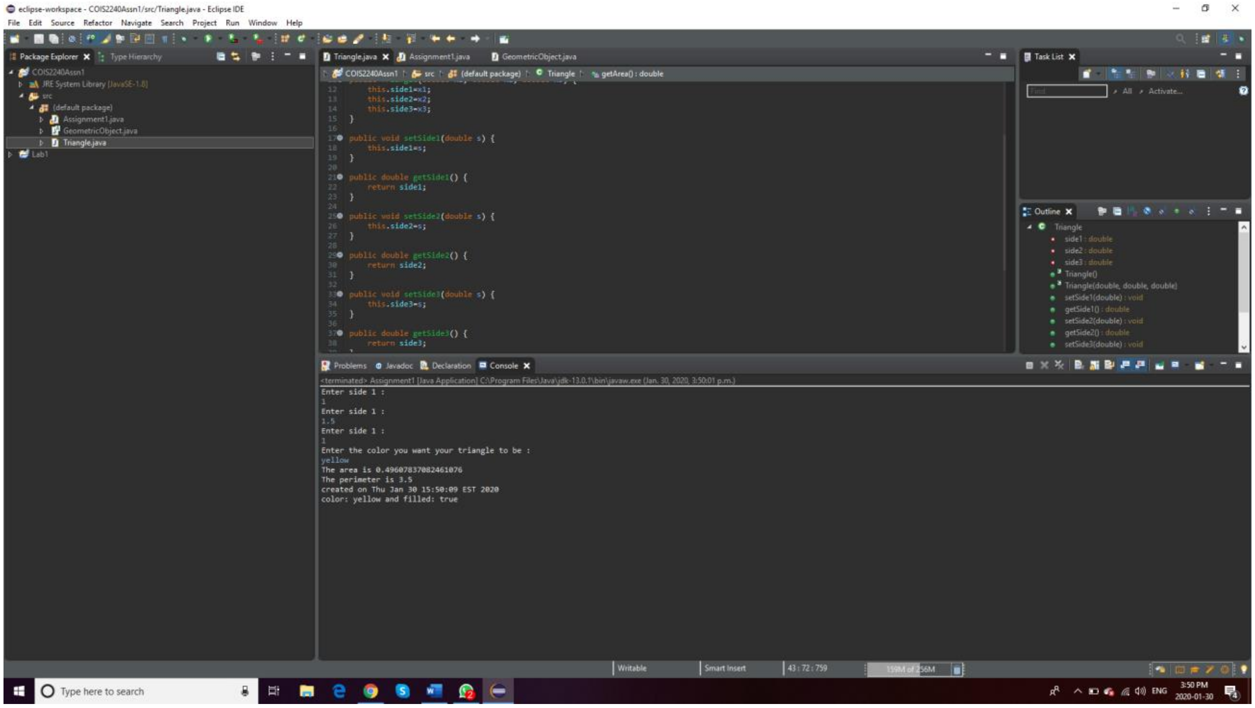
• The get and set methods for all three data fields.

• A method named getArea() that returns the area of this triangle.

• A method named getPerimeter() that returns the perimeter of this triangle.

• A method named toString() that returns a string description for the triangle. The toString() method is implemented as follows:

return "Triangle: side1 = " + side1 + " side2 = " + side2 +   " side3 = " + side3;



```
        this.side1=x1;
        this.side2=x2;
        this.side3=x3;
    }

    public void setSide1(double s) {
        this.side1=s;
    }

    public double getSide1() {
        return side1;
    }

    public void setSide2(double s) {
        this.side2=s;
    }

    public double getSide2() {
        return side2;
    }

    public void setSide3(double s) {
        this.side3=s;
    }

    public double getSide3() {
        return side3;
```

```
<terminated> Assignment1 [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Jan. 30, 2020, 3:50:01 p.m.)
Enter side 1 :
1
Enter side 1 :
1.5
Enter side 1 :
1
Enter the color you want your triangle to be :
yellow
The area is 0.49607837082461076
The perimeter is 3.5
created on Thu Jan 30 15:50:09 EST 2020
color: yellow and filled: true
```

The program runs.

We ask the user for first side, second side, third side and the color and print it out with area, perimeter and if or not it’s filled.