Name: Arsalan Shaikh

Roll No:31

Class: S.Y.B.C.A

Division: ’A’

Semester: IV

Year: 2021-2022

Dnyanprassarak Mandal’s

College and Research Centre, Assagao- Goa.

CAC-115: CASE TOOLS LABORATORY

Practical Journal

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Session No.** | **Title** | **Date** | **Page No.** | **Teacher’s Signature** |
| 1. | Code Documentation Tool- Javadoc  Session 1- Practice Java Programs | 09/03/2022 | 2 |  |
| 2. | Code Documentation Tool- Javadoc  Session 2- Java Programs & Java Doc | 09/03/2022 | 2-12 |  |
| 3. | Code Documentation Tool- Javadoc  Session 3- Pair Programming | 23/03/2022 | 12-46 |  |
| 4. | Session 4-GIT | 31/03/2022 | 47-54 |  |
| 5. | Session 5-GIT | 07/04/2022 | 54-58 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**SESSION 1: Code Documentation using Javadoc**

**Objective:** To learn to document code and generate documentation using the Javadoc tool.

**Date**: 09/03/2022

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Implement the following java programs in the Eclipse IDE:

1. Print any string in Java (e.g. "Hello World!").

Code:

/\*\*

\* <h1>Hello World</h1>

\* The hello world program implements an application that simply displays "Hello World" on output.

\* **@author** Arsalan

\* **@version** 1.0

\*

\*/

**public** **class** HelloWorld

{

**public** **static** **void** main(String[] args)

{

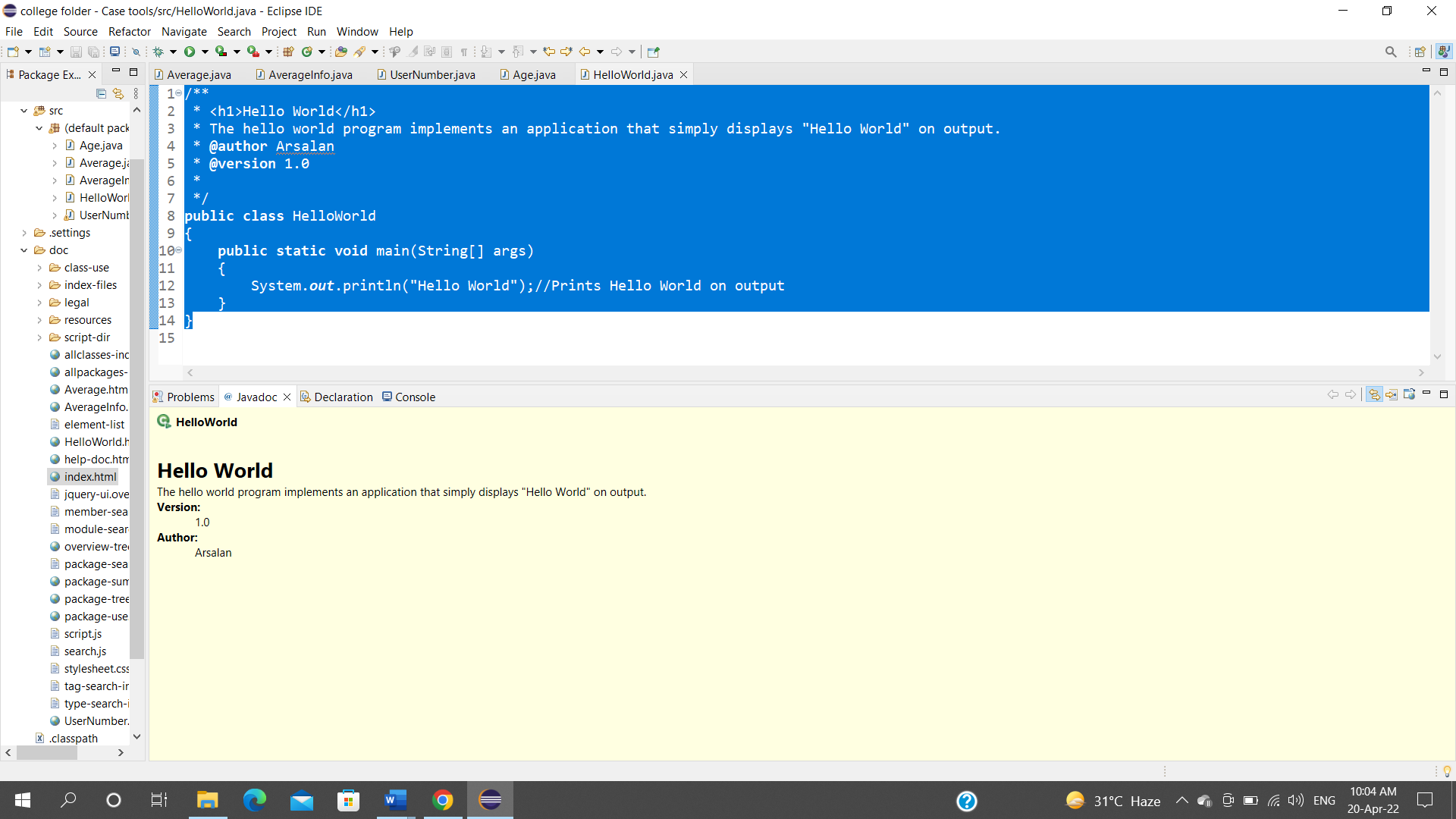
System.***out***.println("Hello World");//Prints Hello World on output

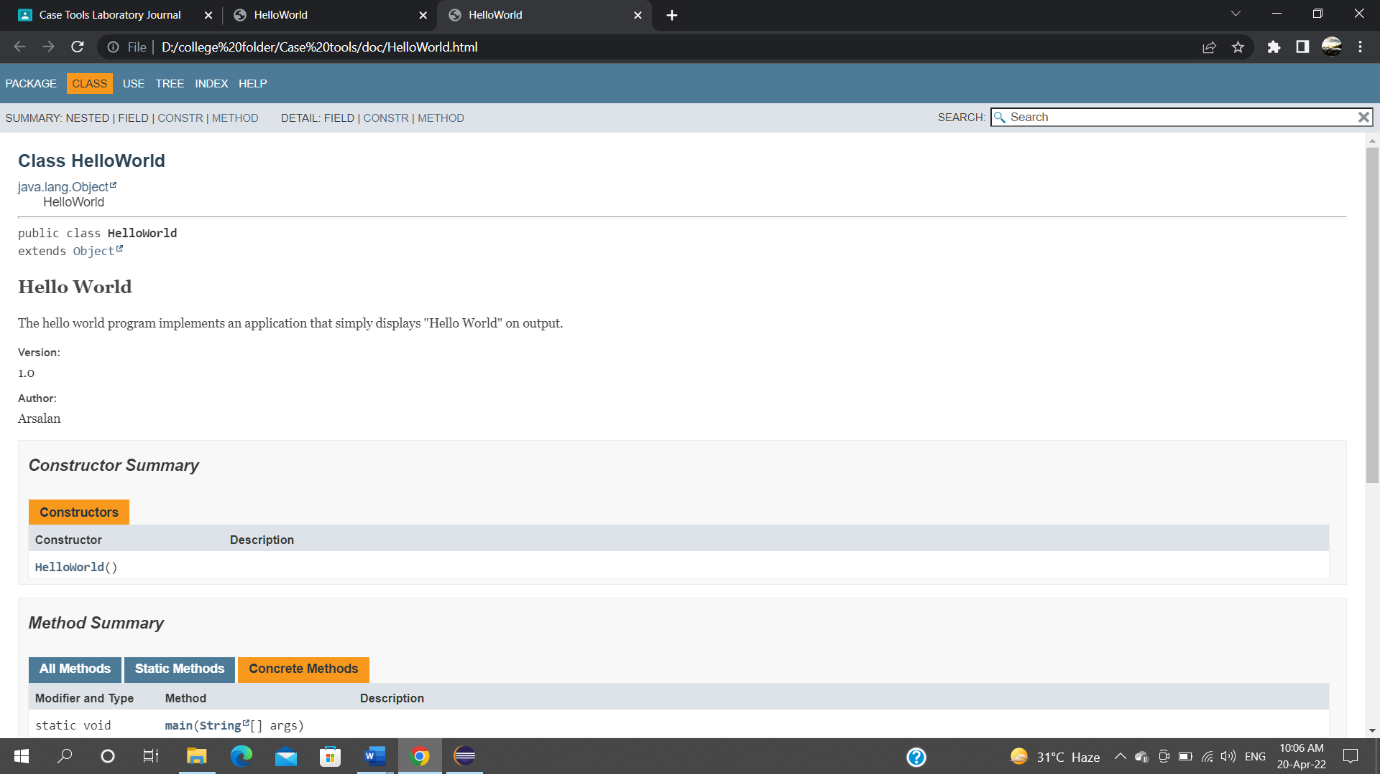
}

}

Output

Hello World





1. Add two numbers and find their average.

**public** **class** Average

{

/\*\*

\* Class Average

\* **@param** num1

\* **@param** num2

\* **@param** average

\*/

**int** num1,num2,average;

/\*\*\*

\*

\* **@return** Num1

\*/

**public** **int** getNum1() {

**return** num1;

}

**public** **void** setNum1(**int** num1) {

**this**.num1 = num1;

}

/\*\*\*

\*

\* **@return** Num2

\*/

**public** **int** getNum2() {

**return** num2;

}

**public** **void** setNum2(**int** num2) {

**this**.num2 = num2;

}

/\*\*\*

\*

\* **@return** average

\*/

**public** **int** getAverage() {

average=(num1+num2)/2;

System.***out***.println("Average="+average);

**Return2** average;

}

/\*\*\*

\*

\* **@param** average

\*/

**public** **void** setAverage(**int** average) {

**this**.average = average;

}

}

/\*\*\*

\*

\* **@author** Arsalan

\* **@version** 1.0

\* <h1> Program to find Average of two numbers </h1>

\* <p>This is the main class which makes an object of the Average class and calls the methods to find the average

\* </p>

\* \*/

**public** **class** AverageMain

{

**public** **static** **void** main(String []args)

{

Average av=**new** Average();

av.setNum1(10);/\*Input First number \*/

av.getNum1();

av.setNum2(40);/\*Input second number\*/

av.getNum2();

av.getAverage();

}

}

Output:

Average=25

1. Accept a number from the user and display it.

**import** java.util.Scanner;

**public** **class** displaynum

{

/\*\*\*

\* **@param** num1

\*/

**int** num1;

Scanner sc=**new** Scanner(System.***in***);

/\*\*\*

\*

\* **@return** num1

\*/

**public** **int** getNum1() {

**return** num1;

}

**public** **void** setNum1(**int** num1) {

**this**.num1 = num1;

}

/\*\*\*

\* Display method to print the number

\*/

**void** display()

{

System.***out***.println("The number is " +num1);

}

}

**import** java.util.Scanner;

**public** **class** displaynumMain

/\*\*\*

\* **@author** Arsalan

\* Program to Accept a number from user and display it

\*/

{

**public** **static** **void** main(String[] args) {

**int** num1;

displaynum dis=**new** displaynum();

Scanner sc=**new** Scanner(System.***in***);

dis.getNum1();

System.***out***.println("Enter the number");

num1=sc.nextInt();

dis.setNum1(num1);

dis.display();

}

}

Output:

Enter the number

2

The number is 2

1. Accept three numbers from the user and find the greatest.

**import** java.util.Scanner;

**public** **class** great

{

Scanner sc=**new** Scanner(System.***in***);

/\*\*\*

\* **@param** num1

\* **@param** num2

\* **@param** num3

\* **@param** great

\*/

/\*\*\*

\* **@return** num2

\*/

**int** num1,num2,num3,great;

**public** **int** getNum2() {

**return** num2;

}

**public** **void** setNum2(**int** num2) {

**this**.num2 = num2;

}

/\*\*\*

\*

\* **@return** num3

\*/

**public** **int** getNum3() {

**return** num3;

}

**public** **void** setNum3(**int** num3) {

**this**.num3 = num3;

}

/\*\*\*

\*

\* **@return** greatest number comparing 3 numbers

\*/

**public** **int** getGreat() {

**if** (num1<num2 && num2>num3) {

System.***out***.println("Num is greater is "+num2);

}

**else** **if**(num2<num3 && num3>num1) {

System.***out***.println("Num greater is "+num3);

}

**else**

{

System.***out***.println("Num greater is "+num1);

}

**return** great;

}

**public** **void** setGreat(**int** great) {

**this**.great = great;

}

/\*\*\*

\*

\* **@return** num1

\*/

**public** **int** getNum1() {

**return** num1;

}

**public** **void** setNum1(**int** num1) {

**this**.num1 = num1;

}

}

**import** java.util.Scanner;

**public** **class** greatMain

/\*\*\*

\* **@author** Arsalan

\* Program to accept 3 numbers from the user and find the greatest

\*/

{

**public** **static** **void** main(String[] args) {

**int** num1;

great dis=**new** great();

Scanner sc=**new** Scanner(System.***in***);

dis.getNum1();

System.***out***.println("Enter the first number");

num1=sc.nextInt();

dis.setNum1(num1);

dis.getNum2();

System.***out***.println("Enter the second number");

**int** num2 = sc.nextInt();

dis.setNum2(num2);

dis.getNum3();

System.***out***.println("Enter the third number");

**int** num3 = sc.nextInt();

dis.setNum3(num3);

dis.getGreat();

}

}

Output:

Enter the first number

6

Enter the second number

10

Enter the third number

3

Num is greater is 10

1. Create a class Product with data members id, name and cost, and constructors to initialize the data.

/\*\*

\* **@author** Arsalan

\* This class has parameterized constructor

\*/

**public** **class** product

{

**int** id,cost;

String name;

/\*\*\*

\*

\* **@param** x

\* **@param** y

\* **@param** z

\*/

//Parameterized constructor

product(**int** x ,**int** y,String z)

{

id=x;

cost=y;

name=z;

}

/\*\*\*

\* Displays the result

\*/

**void** display()

{

System.***out***.println(" Product details are : "+id +" "+cost +" " +name);

}

}

/\*\*

\* **@author** Arsalan

\* **@version** 1.0.2

\* <h1>This is the main class which calls class product </h1>

\*

\*/

**public** **class** Pmain {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args)

{

product a1 =**new** product(110,500,"Lipstick");

a1.display();

}

}

Output:

Product details are : 110 500 Lipstick

1. Demonstrate single inheritance using two classes Floor and Room.

/\*\*

\* <h1>This is class Floor which will find the Area of the floor</h1>

\*

\*/

**public** **class** floor {

/\*\*

\* **@param** length

\* **@param** breadth

\* **@param** area

\*/

**int** length,breadth,area;

**public** **int** getLength() {

**return** length;

}

**public** **void** setLength(**int** length) {

**this**.length = length;

}

**public** **int** getBreadth() {

**return** breadth;

}

**public** **void** setBreadth(**int** breadth) {

**this**.breadth = breadth;

}

**public** **int** getArea() {

area=length\*breadth;

System.***out***.println("The area of the floor is "+area);

**return** area;

}

**public** **void** setArea(**int** area) {

**this**.area = area;

}

}

/\*\*

\* <h1>This is class Room which extends class Floor</h1>

\* This Class states the room type

\*

\*/

**public** **class** room **extends** floor{

/\*\*

\* **@param** number

\* **@param** Roomtype

\*/

**int** number; String Roomtype;

**public** **int** getNumber() {

**return** number;

}

**public** **void** setNumber(**int** number) {

**this**.number = number;

}

**public** String getRoomtype() {

**return** Roomtype;

}

**public** **void** setRoomtype(String roomtype) {

Roomtype = roomtype;

}

**void** display()

{

System.***out***.println("Room type is"+Roomtype);

}

}

/\*\*

\* **@author** Arsalan

\* **@version** 1.0.2

\* <h1>This is the main class which calls class Room </h1>

\* The parent class is floor and child class is room

\*/

**public** **class** singleMain {

/\*\*

\* Main method to call class Room which is the child class

\*\*/

**public** **static** **void** main(String[] args)

{

room a=**new** room();

a.setLength(3);

a.setBreadth(2);

a.setRoomtype(" classic");

a.display();

a.getArea();

}

}

Output:

Room type is classic

The area of the floor is 6

**SESSION 2: Code Documentation using Javadoc\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:** To implement Java programs in Eclipse and generate appropriate documentation for the same using Javadoc

**Date**:09/03/2022

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Find the average marks of a student scored in Math, Science and English. (Assume each subject is out of 50 marks).**

**Source:**

/\*\*

\* This program is to calculate the average of 3 subjects of a student

\*

\* **@author** Arsalan

\* **@version** 1.0.0

\*/

**public** **class** AverageMarks {

**float** math;

**float** science;

**float** english;

/\*\*

\* This method is used to return the marks obtained in Math

\*

\* **@return** math

\*/

**public** **float** getMath() {

**return** math;

}

/\*\*

\* This method is used to accept the marks obtained in Math

\*

\* **@param** math

\*/

**public** **void** setMath(**float** math) {

**this**.math = math;

}

/\*\*

\* This method is used to return the marks obtained in Science

\*

\* **@return** science

\*/

**public** **float** getScience() {

**return** science;

}

/\*\*

\* This method is used to accept the marks obtained in Science

\*

\* **@param** science

\*/

**public** **void** setScience(**float** science) {

**this**.science = science;

}

/\*\*

\* This method is used to return the marks obtained in English

\*

\* **@return**

\*/

**public** **float** getEnglish() {

**return** english;

}

/\*\*

\* This method is used to accept the marks obtained in English

\*

\* **@param** english

\*/

**public** **void** setEnglish(**float** english) {

**this**.english = english;

}

}

**public** **class** StudentMain {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

AverageMarks a = **new** AverageMarks();

a.setEnglish(40);

a.setMath(45);

a.setScience(49);

**float** avg = (a.getEnglish()+a.getMath()+a.getScience()) / 3;

System.***out***.println("Average = " + avg);

}

}

**Output:**

Average = 44.666668

1. **Accept the ages of two people and find who is the younger amongst them.**

**Source Code:**

/\*\*

\* This program is to find the youngest of 2 people

\*

\* **@author** Arsalan

\* **@version** 1.0.0

\*/

**public** **class** YoungestPerson {

**int** person1;

**int** person2;

/\*\*

\* This method returns the age of Person 1

\*

\* **@return** the person1

\*/

**public** **int** getPerson1() {

**return** person1;

}

/\*\*

\* This method declares the age of Person 1

\*

\* **@param** person1 the person1 to set

\*/

**public** **void** setPerson1(**int** person1) {

**this**.person1 = person1;

}

/\*\*

\* This method returns the age of Person 2

\*

\* **@return** the person2

\*/

**public** **int** getPerson2() {

**return** person2;

}

/\*\*

\* This method declares the age of Person 2

\*

\* **@param** person2 the person2 to set

\*/

**public** **void** setPerson2(**int** person2) {

**this**.person2 = person2;

}

}

/\*\*

\* **@author** Arsalan

\*

\*/

**public** **class** YoungestMain {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

YoungestPerson x = **new** YoungestPerson();

x.setPerson1(18);

x.setPerson2(18);

*youngestPerson*(x.getPerson1(), x.getPerson2());

}

**public** **static** **void** youngestPerson(**int** a, **int** b) {

**if** (a == b) {

System.***out***.println("Both are of the same age");

**return**;

}

**else** **if** (a < b) {

System.***out***.println("Person A with " + a + " age is younger");

}

**else** {

System.***out***.println("Person B with " + b + " age is younger");

}

}

}

**Output:**

Person A with 18 age is younger

1. **Create a class Item with data members Id and Name and member functions to get and display the data members.**

**Source Code:**

/\*\* This program is to display the id and name of a person

\* **@author** Arsalan

\*

\*/

**public** **class** IdName {

**int** id;

String name;

/\*\*This method returns the ID value

\* **@return** the id

\*/

**public** **int** getId() {

**return** id;

}

/\*\*This method sets the value of id variable

\* **@param** id the id to set

\*/

**public** **void** setId(**int** id) {

**this**.id = id;

}

/\*\*This method returns the Name value

\* **@return** the name

\*/

**public** String getName() {

**return** name;

}

/\*\*This method sets the value of name variable

\* **@param** name the name to set

\*/

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **void** displayData() {

System.***out***.println("ID = " + getId());

System.***out***.println("Name = " + getName());

}

}

/\*\*

\* **@author** Arsalan

\*

\*/

**public** **class** IdNameMain {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

IdName i = **new** IdName();

i.setId(20010);

i.setName("Bruce Wayne");

i.displayData();

}

}

**Output:**

ID = 20010

Name = Bruce Wayne

1. **Create a class Student with data members Roll No, Age and Name and member functions to get and display the data members from the keyboard.**

**Source Code:**

/\*\*

\* **@author** Arsalan

\*

\*/

**public** **class** Student {

**int** rollNo;

**int** age;

String name;

/\*\*

\* **@return** the rollNo

\*/

**public** **int** getRollNo() {

**return** rollNo;

}

/\*\*

\* **@param** rollNo the rollNo to set

\*/

**public** **void** setRollNo(**int** rollNo) {

**this**.rollNo = rollNo;

}

/\*\*

\* **@return** the age

\*/

**public** **int** getAge() {

**return** age;

}

/\*\*

\* **@param** age the age to set

\*/

**public** **void** setAge(**int** age) {

**this**.age = age;

}

/\*\*

\* **@return** the name

\*/

**public** String getName() {

**return** name;

}

/\*\*

\* **@param** name the name to set

\*/

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **void** displayData() {

System.***out***.println("Name: " + getName());

System.***out***.println("Roll No: " + getRollNo());

System.***out***.println("Age: " + getAge());

}

}

**import** java.util.Scanner;

/\*\*

\* **@author** Arsalan

\*

\*/

**public** **class** StudentInfoMain {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Student s = **new** Student();

Scanner input = **new** Scanner(System.***in***);

String name = input.nextLine();

**int** rollNo = input.nextInt();

**int** age = input.nextInt();

s.setName(name);

s.setRollNo(rollNo);

s.setAge(age);

s.displayData();

}

}

**Output:**

Ajay Devgn

20

54

Name: Ajay Devgn

Roll No: 20

Age: 54

1. **Create a class Cars with data members Name, Color and Cost and member functions to get and display the data members. Implement the above for two car objects.**

**Source Code:**

**import** java.util.jar.Attributes.Name;

**public** **class** Car {

String name;

String color;

**int** cost;

/\*\*

\* **@return** the name

\*/

**public** String getName() {

**return** name;

}

/\*\*

\* **@param** name the name to set

\*/

**public** **void** setName(String name) {

**this**.name = name;

}

/\*\*

\* **@return** the color

\*/

**public** String getColor() {

**return** color;

}

/\*\*

\* **@param** color the color to set

\*/

**public** **void** setColor(String color) {

**this**.color = color;

}

/\*\*

\* **@return** the cost

\*/

**public** **int** getCost() {

**return** cost;

}

/\*\*

\* **@param** cost the cost to set

\*/

**public** **void** setCost(**int** cost) {

**this**.cost = cost;

}

**public** **void** displayData() {

System.***out***.println("Name: " + **this**.name);

System.***out***.println("Color: " + **this**.color);

System.***out***.println("Cost: " + **this**.cost + "Rs/-");

}

}

**public** **class** CarMain {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Car c1 = **new** Car();

c1.setName("Rolls Royce");

c1.setColor("Black");

c1.setCost(50000000);

c1.displayData();

System.***out***.println();

Car c2 = **new** Car();

c2.setName("BMW X1");

c2.setColor("White");

c2.setCost(3980000);

c2.displayData();

}

}

**Output:**

Name: Rolls Royce

Color: Black

Cost: 50000000Rs/-

Name: BMW X1

Color: White

Cost: 3980000Rs/-

1. **Calculate Area of Circle, Rectangle, Square and Triangle using Function overloading. [Area of circle = ; Area of Rectangle = length x breadth; Area of Square = side2; Area of Triangle = ½ x base x height]**

**Source Code:**

/\*\*

\* **@author** Arsalan

\* Calculate Area of Circle, Rectangle, Square and Triangle using

\* Function overloading. [Area of circle = ; Area of Rectangle = length

\* x breadth; Area of Square = side2; Area of Triangle = Â½ x base x

\* height]

\*/

**public** **class** Area {

/\*\*

\* This method returns the area of a circle

\*

\* **@param** r

\* **@return**

\*/

**public** Double area(Double r) {

**return** 3.14 \* r \* r;

}

/\*\*

\* This method overrides the area method and returns the area of rectangle

\*

\* **@param** l

\* **@param** b

\* **@return**

\*/

**public** **int** area(**int** l, **int** b) {

**return** l \* b;

}

/\*\*

\* This method overrides the area method and returns the area of square

\*

\* **@param** s

\* **@return**

\*/

**public** **int** area(**int** s) {

**return** s \* s;

}

/\*\*

\* This method overrides the area method and returns the area of triangle

\*

\* **@param** b

\* **@param** h

\* **@return**

\*/

**public** **double** area(**double** b, **double** h) {

**return** (0.5 \* b \* h);

}

}

/\*\*

\* **@author** Arsalan

\*

\*/

**public** **class** AreaMain {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Area a = **new** Area();

**double** rad = 2.5;

**int** len = 3, bre = 7;

**int** sid = 5;

**double** bas = 2.5, hei = 5.5;

// Area of a Circle

System.***out***.println("Radius" + rad);

System.***out***.println("Area (Circle) = " + a.area(rad));

System.***out***.println();

// Area of Rectangle

System.***out***.println("Length = " + len + "\nBreadth = " + bre);

System.***out***.println("Area (Rectangle) = " + a.area(3, 7));

System.***out***.println();

// Area of Square

System.***out***.println("Side = " + sid);

System.***out***.println("Area (Square) = " + a.area(5));

System.***out***.println();

// Area of Triangle

System.***out***.println("Base = " + bas + "\nHeight = " + hei);

System.***out***.println("Area (Triangle) = " + a.area(2.5, 5.5));

System.***out***.println();

}

}

**Output:**

Radius2.5

Area (Circle) = 19.625

Length = 3

Breadth = 7

Area (Rectangle) = 21

Side = 5

Area (Square) = 25

Base = 2.5

Height = 5.5

Area (Triangle) = 6.875

1. **To create a class Product with data members id, name and cost and constructors to initialize the data members of the class**

**Source Code:**

/\*\*

\* **@author** Arsalan

\* 7. To create a class Product with data members id, name and cost and

\* constructors to initialize the data members of the class

\*/

**public** **class** Product {

**int** id;

String name;

**int** cost;

/\*\*

\* This is the class constructor

\*

\* **@param** id

\* **@param** name

\* **@param** cost

\*/

**public** Product(**int** id, String name, **int** cost) {

**this**.id = id;

**this**.name = name;

**this**.cost = cost;

}

**public** **void** displayData() {

System.***out***.println("ID: " + **this**.id);

System.***out***.println("Name: " + **this**.name);

System.***out***.println("cost: " + **this**.cost + "/-");

}

}

/\*\*

\* **@author** abhishek

\*

\*/

**public** **class** ProductMain {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Product phone = **new** Product(1463, "Mi A3", 13999);

phone.displayData();

}

}

**Output:**

ID: 1463

Name: Mi A3

cost: 13999/-

1. **Create a class a vehicle having data members brand, name, cost and roadTax and member functions to set and retrieve the values of data members.Also Write a method calculateRoadTax() which will calculate roadTax depending on the option selected by the user i.e. 1 for bike , 2 for car and 3 for truck. if option selected is 1 then roadTax = 500, if option selected is 2 then roadTax = 1000 ,if option selected is 3 then roadTax = 1500. Create an interface Amount which will have a data member govtTax having a constant value of 2000 and a member function to calculate the final amount. Create another class called Result which will inherit from the two above classes and calculate the final amount as the sum of govtTax, cost and roadTax the values of which should be inherited from the above two classes.**

**Source Code:**

**import** java.util.Scanner;

/\*\*

\* **@author** Arsalan

\*

\*/

**public** **class** Vehicle {

String brand, name;

**int** cost, roadTax;

/\*\*

\* **@return** the brand

\*/

**public** String getBrand() {

**return** brand;

}

/\*\*

\* **@param** brand the brand to set

\*/

**public** **void** setBrand(String brand) {

**this**.brand = brand;

}

/\*\*

\* **@return** the name

\*/

**public** String getName() {

**return** name;

}

/\*\*

\* **@param** name the name to set

\*/

**public** **void** setName(String name) {

**this**.name = name;

}

/\*\*

\* **@return** the cost

\*/

**public** **int** getCost() {

**return** cost;

}

/\*\*

\* **@param** cost the cost to set

\*/

**public** **void** setCost(**int** cost) {

**this**.cost = cost;

}

/\*\*

\* **@return** the roadTax

\*/

**public** **int** getRoadTax() {

**return** roadTax;

}

/\*\*

\* **@param** roadTax the roadTax to set

\*/

**public** **void** setRoadTax(**int** roadTax) {

**this**.roadTax = roadTax;

}

**public** **void** calcRoadTax() {

Scanner input = **new** Scanner(System.***in***);

System.***out***.println("1. Bike");

System.***out***.println("2. Car");

System.***out***.println("3. Truck");

**int** response = input.nextInt();

**switch** (response) {

**case** 1:

roadTax = 500;

**break**;

**case** 2:

roadTax = 1000;

**break**;

**case** 3:

roadTax = 1500;

**break**;

**default**:

**break**;

}

}

}

/\*\*

\* **@author** Arsalan

\*

\*/

**public** **interface** Amount {

**final** **int** ***govtTax*** = 2000;

**abstract** **void** finalAmount();

**public** **default** **int** getGovtTax() {

**return** ***govtTax***;

}

}

/\*\*

\* **@author** Arsalan

\*

\*/

**public** **class** Result **extends** Vehicle **implements** Amount {

**int** finalAmountVariable;

**public** **void** finalAmount() {

finalAmountVariable = ***govtTax*** + roadTax + cost;

}

/\*\*

\* **@return** the finalAmountVariable

\*/

**public** **int** getFinalAmountVariable() {

**return** finalAmountVariable;

}

}

/\*\*

\* **@author** Arsalan

\*

\*/

**public** **class** FinalMain {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Result r = **new** Result();

r.setBrand("Mahindra");

r.setName("Scorpio");

r.setCost(1000000);

r.calcRoadTax();

r.finalAmount();

System.***out***.println("Brand: " + r.getBrand());

System.***out***.println("Name: " + r.getName());

System.***out***.println("Cost: " + r.getCost());

System.***out***.println("Road Tax: " + r.getRoadTax());

System.***out***.println("Govt Tax: " + r.getGovtTax());

System.***out***.println("Final Amount: " + r.getFinalAmountVariable());

}

}

**Output:**

1. Bike

2. Car

3. Truck

3

Brand: Mahindra

Name: Scorpio

Cost: 1000000

Road Tax: 1500

Govt Tax: 2000

Final Amount: 1003500

1. **Demonstrate divide by zero Exception.**

**Source Code:**

/\*\*

\* **@author** Arsalan

\* This is a program to show DivideByZero error

\*/

**public** **class** DivideByZero {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n1 = 5;

**int** n2 = 0;

System.***out***.print(n1 / n2);

}

}

**Output:**

Exception in thread "main" java.lang.ArithmeticException: / by zero

at DivideByZero.main(DivideByZero.java:8)

1. **Demonstrate ArrayOutOfBounds Exception.**

**Source Code:**

/\*\*

\* **@author** Arsalan

\* This program is to show the ArrayOutOfBound error;

\*/

**public** **class** ArrayError {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int**[] num = { 1, 2, 3, 4 };

System.***out***.print(num[4]);

}

}

**Output:**

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4

at ArrayError.main(ArrayError.java:13)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**=====================================================**

**Session 2:** Pair Programming

**Objective:** To use pair programming for program                         development

**Date:** 23/03/2022-24/03/2022

**=====================================================**

1. **Implement the following programs with pair programming and provide appropriate documentation for the same:**

#### The Circle Class:

Design a class called *circle* which contains:

* Two private instance variables: radius (of the type double) and colour (of the type String), with default value of 1.0 and "red", respectively.
* Two overloaded constructors - a default constructor with no argument, and a constructor which takes a double argument for radius.
* Two public methods: getRadius() and getArea(), which return the radius and area of this instance, respectively.

**Source Code:**

/\*\*

\* **@author** Arsalan

\*

\*/

**public** **class** Circle {

**double** radius;

String color;

**double** area;

**public** Circle() {

**super**();

**this**.radius = 1.0;

**this**.color = "Red";

setArea(**this**.radius);

}

/\*\*

\* **@param** radius

\* **@param** color

\*/

**public** Circle(**double** radius, String color) {

**this**.radius = radius;

**this**.color = color;

setArea(**this**.radius);

}

/\*\*

\* **@return** the radius

\*/

**public** **double** getRadius() {

**return** radius;

}

/\*\*

\* **@return** the area

\*/

**public** **double** getArea() {

**return** area;

}

/\*\*

\* **@param** area the area to set

\*/

**public** **void** setArea(**double** radius) {

**this**.area = 3.14 \* radius \* radius;

}

/\*\*

\* **@return** the color

\*/

**public** String getColor() {

**return** color;

}

**public** **void** displayData() {

System.***out***.println("Color = " + **this**.color);

System.***out***.println("Radius = " + **this**.radius);

System.***out***.println("Area = " + **this**.area);

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class CircleMain {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Circle c1 = new Circle();

Circle c2 = new Circle(2.14, "Black");

c1.displayData();

System.out.println();

c2.displayData();

}

}

**Output:**

Color = Red

Radius = 1.0

Area = 3.14

Color = Black

Radius = 2.14

Area = 14.379944000000002

* **The Date Class**

Design a Date class that models a calendar date. It contains data members- day [1,31], month [1, 12] and year [1900, 9999] (No input validation needed). Write methods to set all the member functions. Write another member function to print the date in mm/dd/yyyy format with leading zero.

**Source Code:**

/\*\*

\* @author Arsalan

\*

\*/

public class Date {

int day;

int month;

int year;

/\*\*

\* @return the day

\*/

public int getDay() {

return day;

}

/\*\*

\* @param day the day to set

\*/

public void setDay(int day) {

this.day = day;

}

/\*\*

\* @return the month

\*/

public int getMonth() {

return month;

}

/\*\*

\* @param month the month to set

\*/

public void setMonth(int month) {

this.month = month;

}

/\*\*

\* @return the year

\*/

public int getYear() {

return year;

}

/\*\*

\* @param year the year to set

\*/

public void setYear(int year) {

this.year = year;

}

public void displayDate(){

System.out.println(this.month + "/" + this.day +"/" + this.year );

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class DateMain {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Date d = new Date();

d.setDay(16);

d.setMonth(12);

d.setYear(2022);

d.displayDate();

}

}

**Output:**

6/16/2022

* **The Lamp Class**

Create a class named Lamp. It contains a variable isOn and two methods: turnOn() and turnOff. Inside the main class create two objects of the Lamp class- led and halogen. *led.turnOn()* sets the isOn variable to true and prints the output. *halogen.turnOff()* sets the isOn variable to false and prints the output.

**Source Code:**

/\*\*

\* @author Arsalan

\*

\*/

public class Lamp {

boolean IsOn;

public void turnOn() {

this.IsOn = true;

}

public void turnOff() {

this.IsOn = false;

}

public void displayLamp() {

System.out.println(this.IsOn);

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class Lampmain {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Lamp led = new Lamp();

led.turnOn();

Lamp halogen = new Lamp();

halogen.turnOff();

System.out.print("LED is ");

led.displayLamp();

System.out.print("Halogen is ");

halogen.displayLamp();

}

}

**Output:**

LED is false

Halogen is true

* **The Book Class**

Class Book models information of books. A book has

*Fields:*

* An author
* A title
* An ISBN or International Standard Book Number that provides a unique number used by publishers and book stores.

*Constructors and methods:*

* A null constructor.
* A constructor using 3 parameters for an author, title, and ISBN.
* Getters and Setters.

**Source Code:**

/\*\*

\* @author Arsalan

\*

\*/

public class Book {

String author;

String title;

double isbn;

/\*\*

\* @param author

\* @param title

\* @param isbn

\*/

public Book() {

super();

this.author = "null";

this.title = "null";

this.isbn = 0;

}

/\*\*

\* @param author

\* @param title

\* @param isbn

\*/

public Book(String author, String title, double isbn) {

this.author = author;

this.title = title;

this.isbn = isbn;

}

/\*\*

\* @return the author

\*/

public String getAuthor() {

return author;

}

/\*\*

\* @return the title

\*/

public String getTitle() {

return title;

}

/\*\*

\* @return the isbn

\*/

public double getIsbn() {

return isbn;

}

public void displayData() {

System.out.println("author = "+ this.author );

System.out.println("title = "+ this.title );

System.out.println("isbn = "+ this.isbn );

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class BookMain {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Book b1 = new Book();

Book b2 = new Book("chetan bhagat", "400 Days", 12345);

b1.displayData();

b2.displayData();

}

}

**Output:**

author = null

title = null

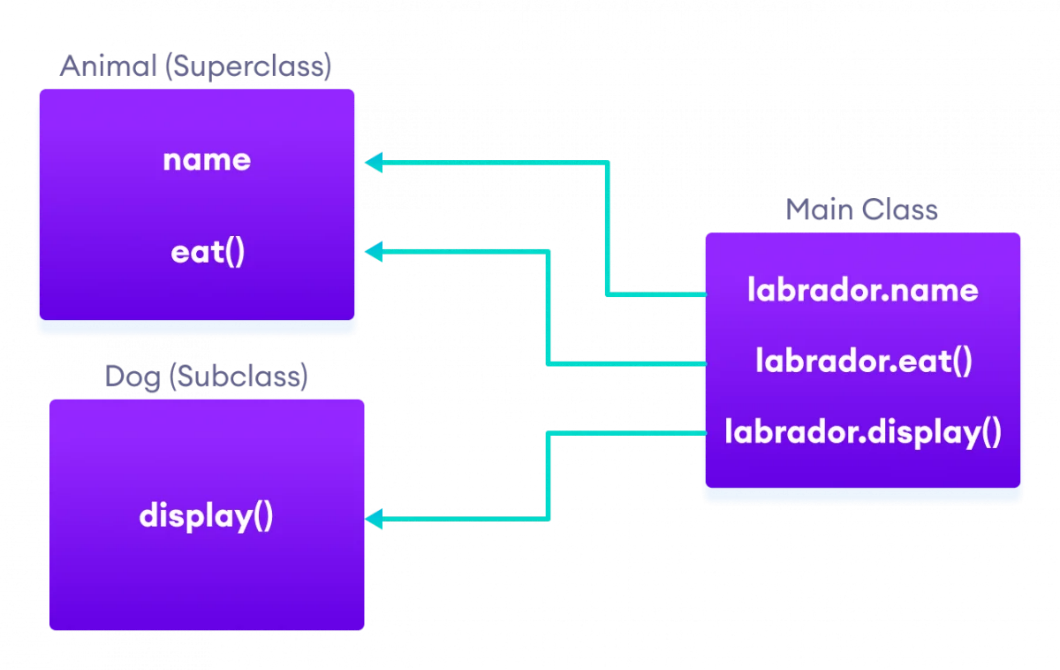
isbn = 0.0

author = chetan bhagat

title = 400 Days

isbn = 12345.0

1. **Implement a Java program for the scenario given below:**



**Source Code:**

/\*\*

\* @author Arsalan

\*

\*/

public class Animal {

String name;

/\*\*

\* @param name

\*/

public Animal(String name) {

this.name = name;

}

public void eat() {

System.out.println("I eat Pedigree");

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class Dog extends Animal {

/\*\*

\* @param name

\*/

public Dog(String name) {

super(name);

// TODO Auto-generated constructor stub

}

public void display() {

System.out.println(name);

eat();

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class DogMain {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Dog labrador = new Dog("Tommy");

labrador.display();

}

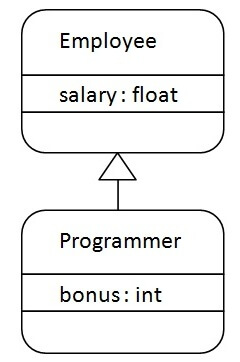
}

**Output:**

Tommy

I eat Pedigree

1. **Implement a Java Program for the scenario given below:**



**Source Code:**

/\*\*

\* @author Arsalan

\*

\*/

public class Employee {

float salary;

/\*\*

\* @param salary

\*/

public Employee(float salary) {

this.salary = salary;

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class Programmer extends Employee{

int bonus;

/\*\*

\* @param salary

\*/

public Programmer(float salary, int bonus) {

super(salary);

this.bonus = bonus;

// TODO Auto-generated constructor stub

}

public void displayData() {

System.out.println("salary = " + salary );

System.out.println("bonus = " + bonus );

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class ProgrammerMain {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Programmer p = new Programmer(25000, 1000);

p.displayData();

}

}

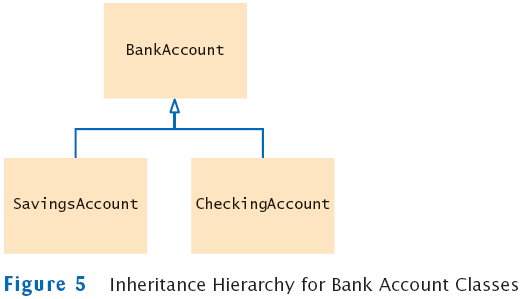
**Output:**

salary = 25000.0

bonus = 1000

1. **Consider a bank that offers its customers the following account types:**
   * **Checking account**: no interest; a small number of free transactions per month, additional transactions are charged a small fee.
   * **Savings account:** earns interest that compounds monthly.

Inheritance hierarchy:



* All bank accounts support the getBalance method.
* All bank accounts support the deposit and *withdraw* methods.
* Checking account needs a method *deductFees*; savings account needs a method *addInterest.*

**Source Code:**

/\*\*

\* @author Arsalan

\*

\*/

public class BankAccount {

double balance;

/\*\*

\* @param balance

\*/

public BankAccount(double balance) {

this.balance = balance;

}

public void deposit(double amount) {

this.balance += amount;

}

public void withraw(double amount) {

this.balance -= amount;

}

/\*\*

\* @return the balance

\*/

public double getBalance() {

return balance;

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class SavingsAccount extends BankAccount {

float interest = 0.01F;

/\*\*

\* @param balance

\*/

public SavingsAccount(double balance) {

super(balance);

// TODO Auto-generated constructor stub

}

public void addInterest() {

balance += balance \* interest;

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class CheckingAccount extends BankAccount{

float deductAmount = 0.0025F;

/\*\*

\* @param balance

\*/

public CheckingAccount(double balance) {

super(balance);

// TODO Auto-generated constructor stub

}

public void deductFees() {

balance -= balance \* deductAmount;

}

}

/\*\*

\* @author Arsalan

\*

\*/

public class AccountMain {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

SavingsAccount sa = new SavingsAccount(50000);

CheckingAccount ca = new CheckingAccount(1000000);

System.out.println("CA = " + ca.getBalance());

System.out.println("SA = " + sa.getBalance());

ca.deductFees();

sa.addInterest();

System.out.println("After adding and deducting");

System.out.println("CA = " + ca.getBalance());

System.out.println("SA = " + sa.getBalance());

}

}

**Output:**

CA = 1000000.0

SA = 50000.0

After adding and deducting

CA = 997500.0000558794

SA = 50499.99998882413

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

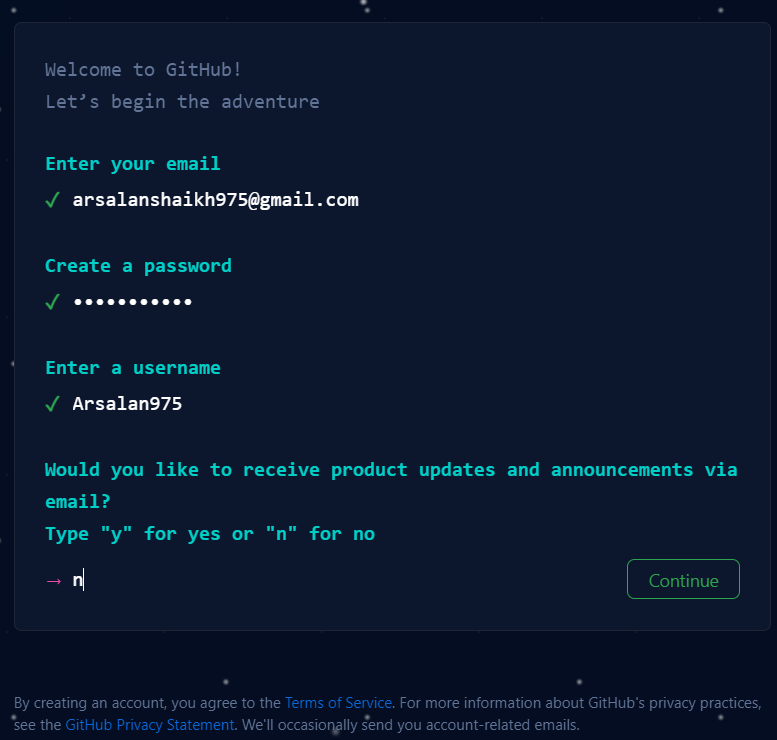
**SESSION 4: GIT\_\_\_\_\_\_\_\_\_\_\_\_\_**

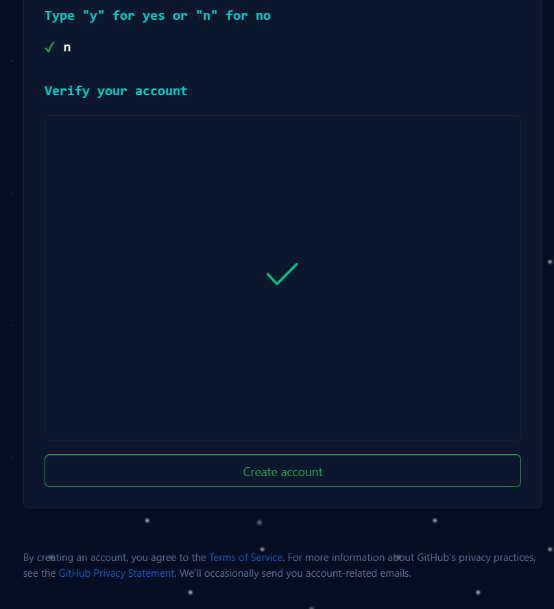
**Objective:** Learn to use centralised repositories and versioning tool.

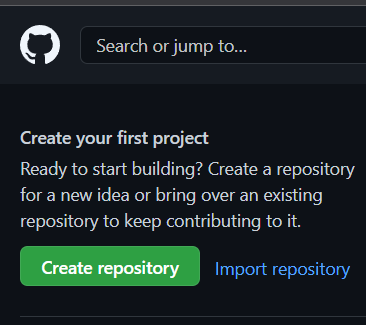
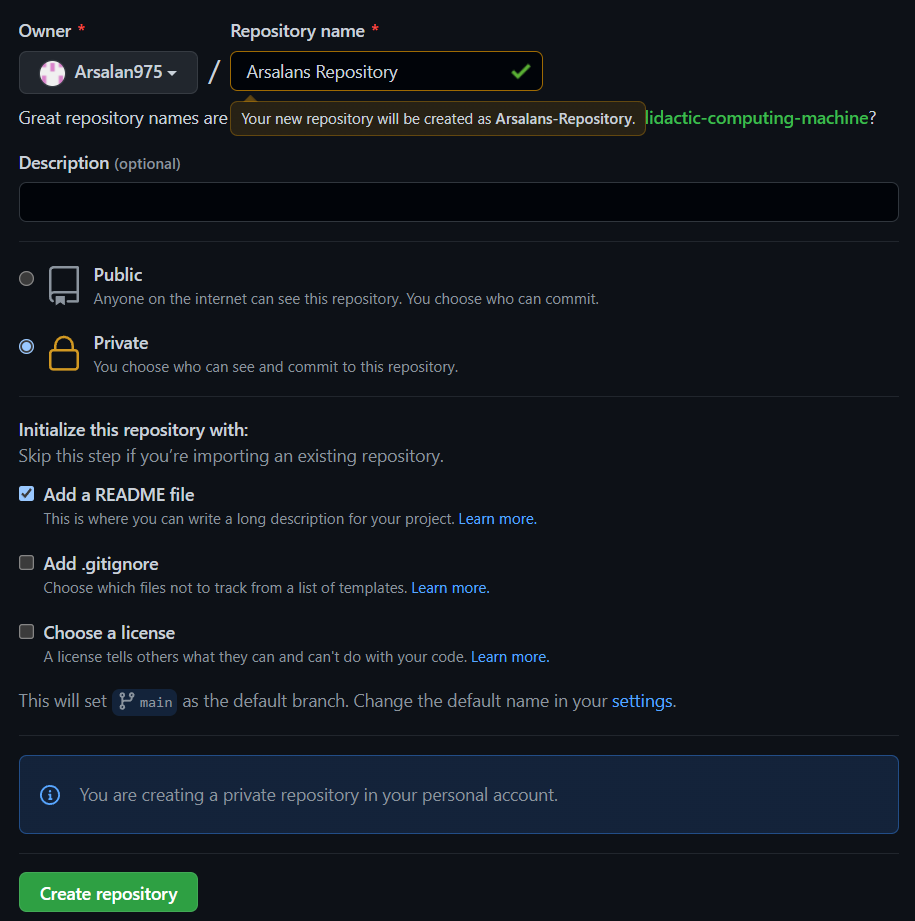
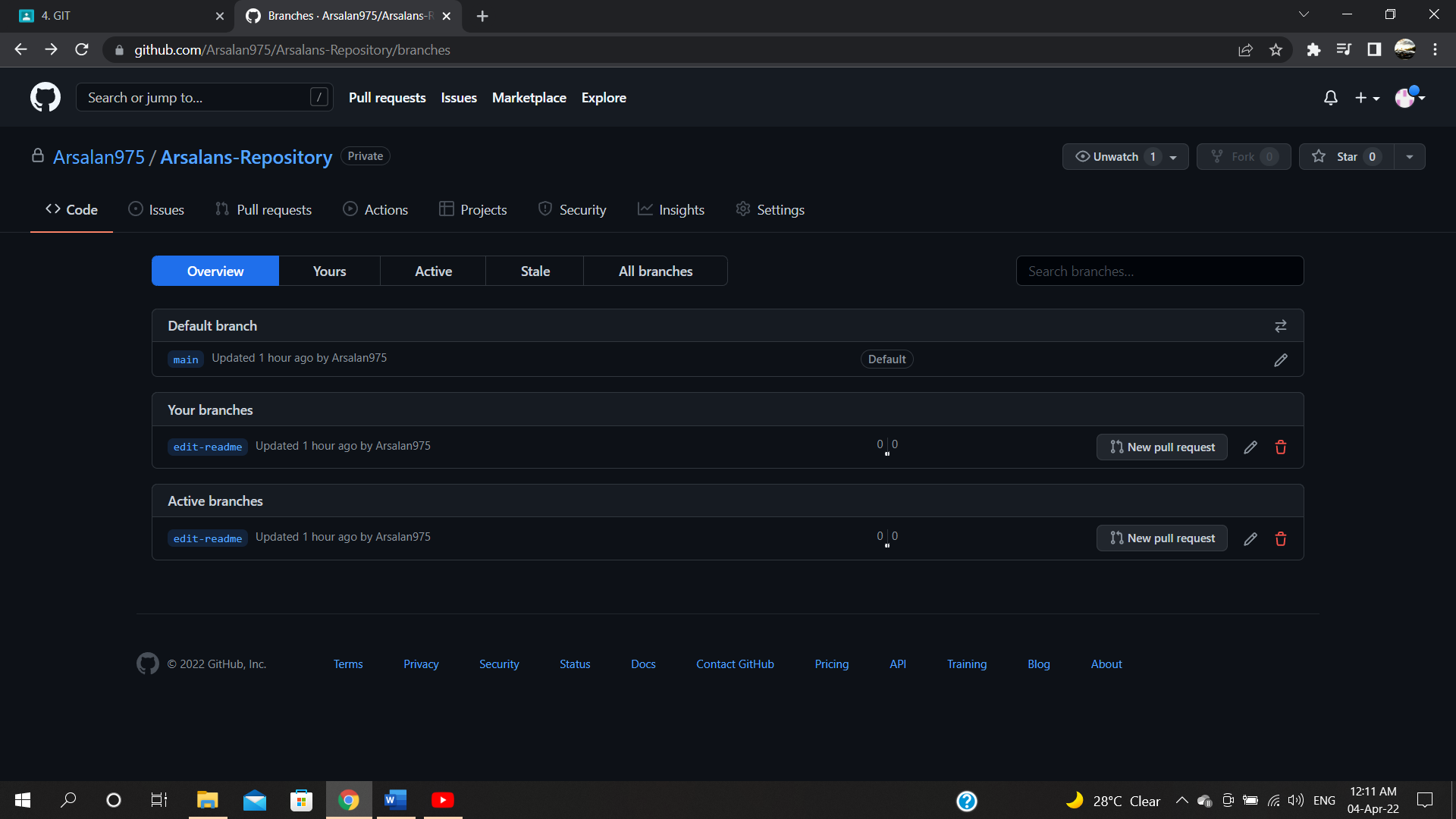
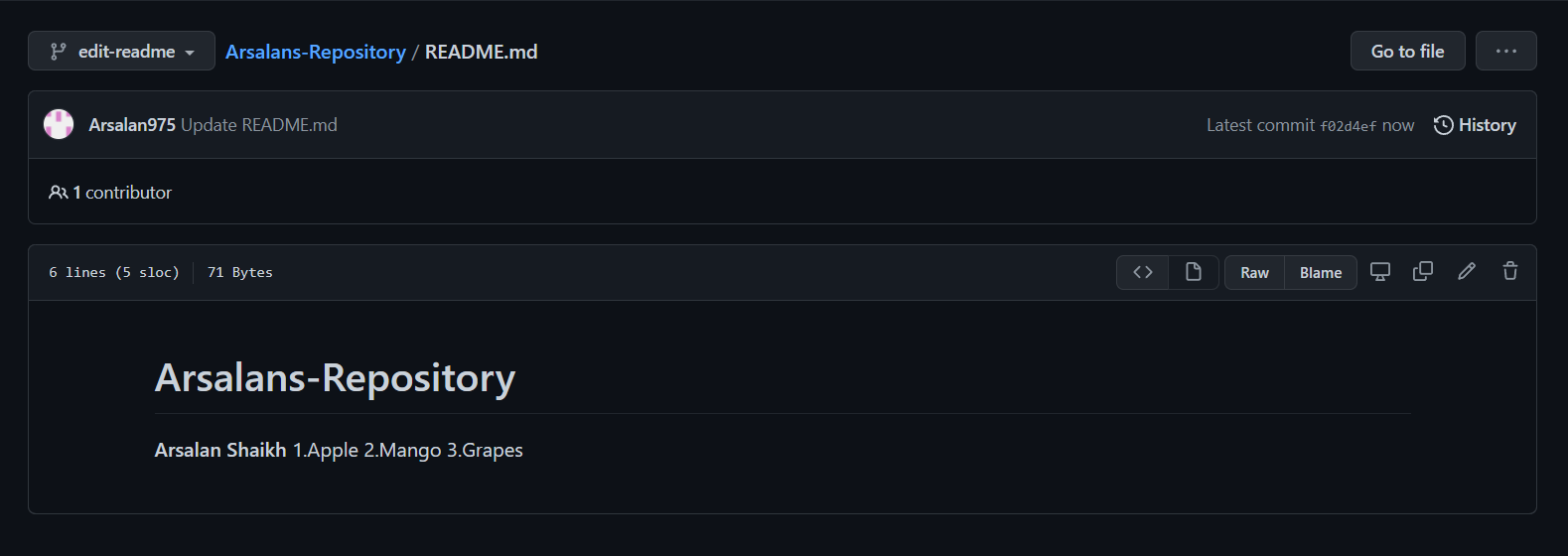
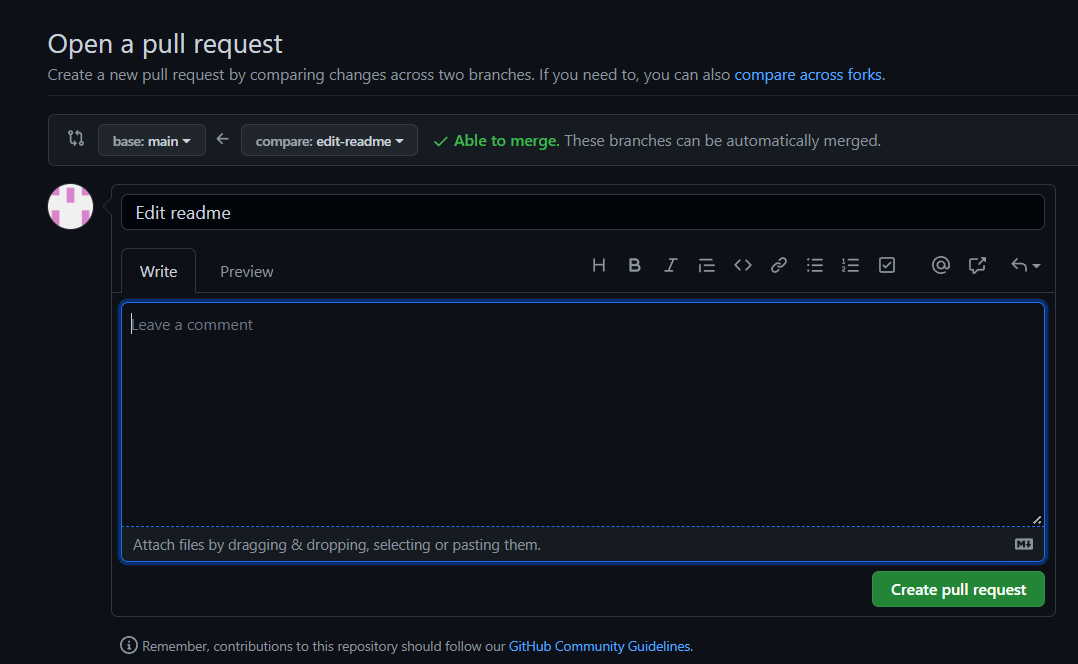
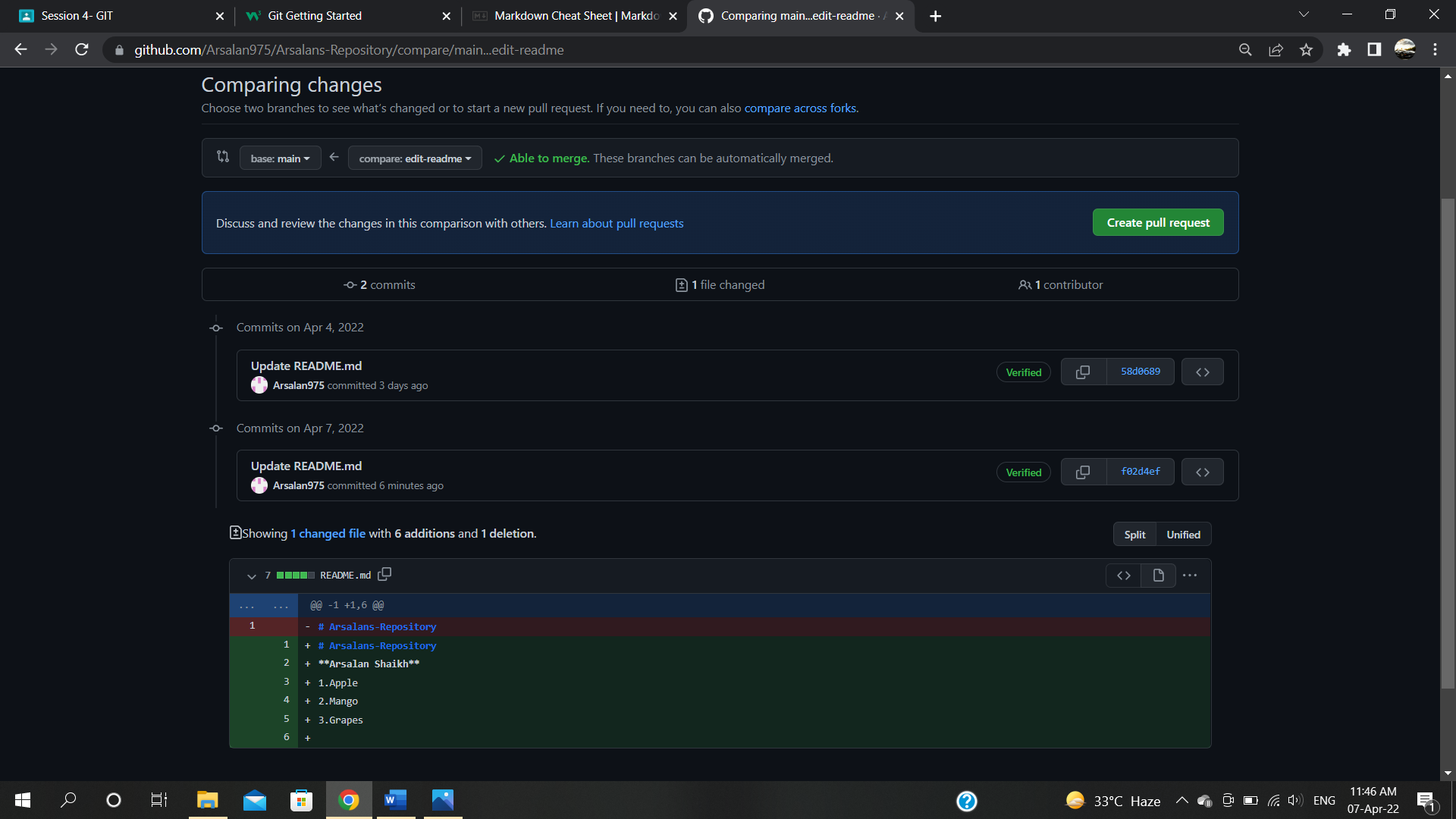
**Date**: 30-03-2022/31-03-2022

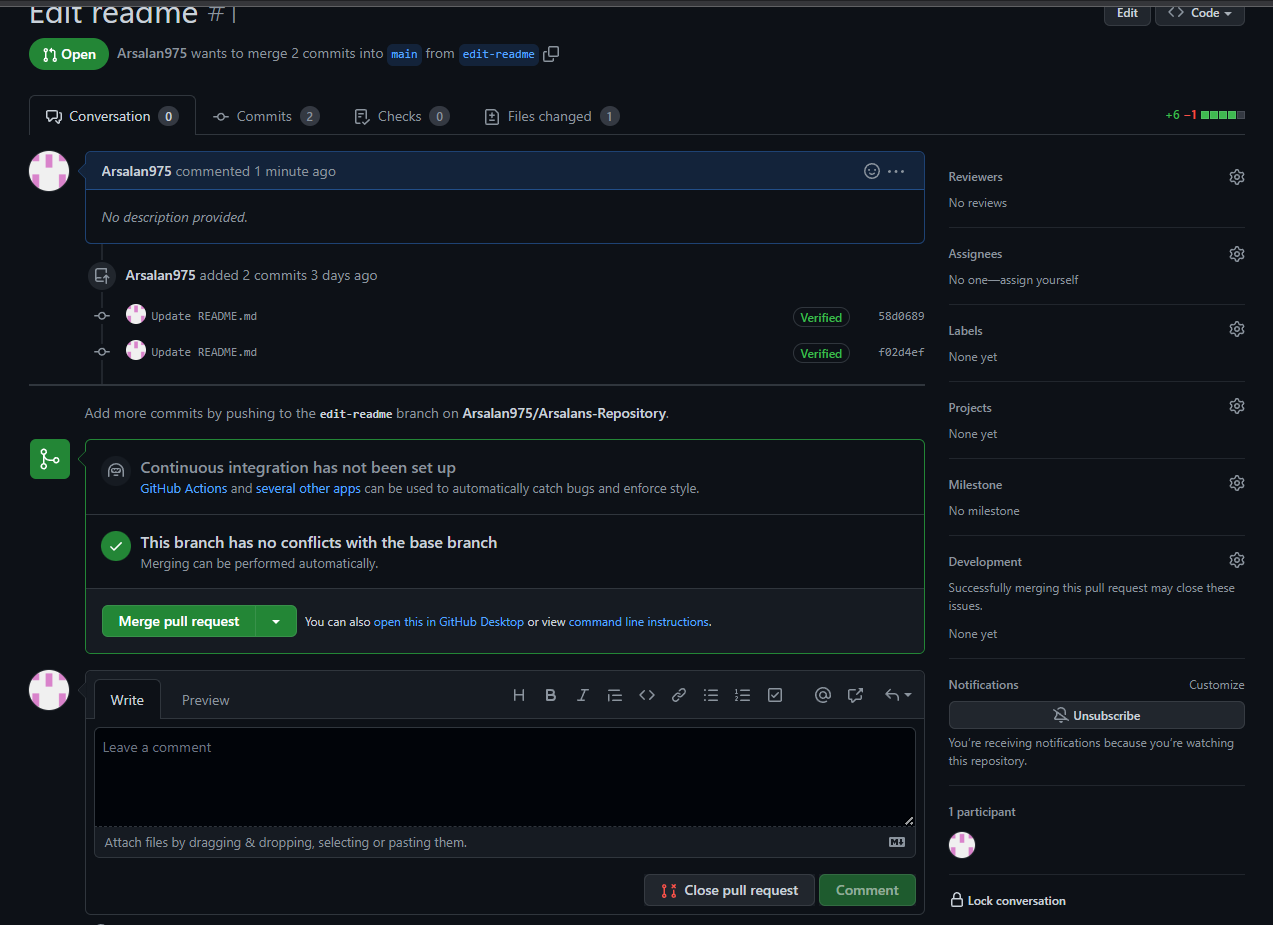
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

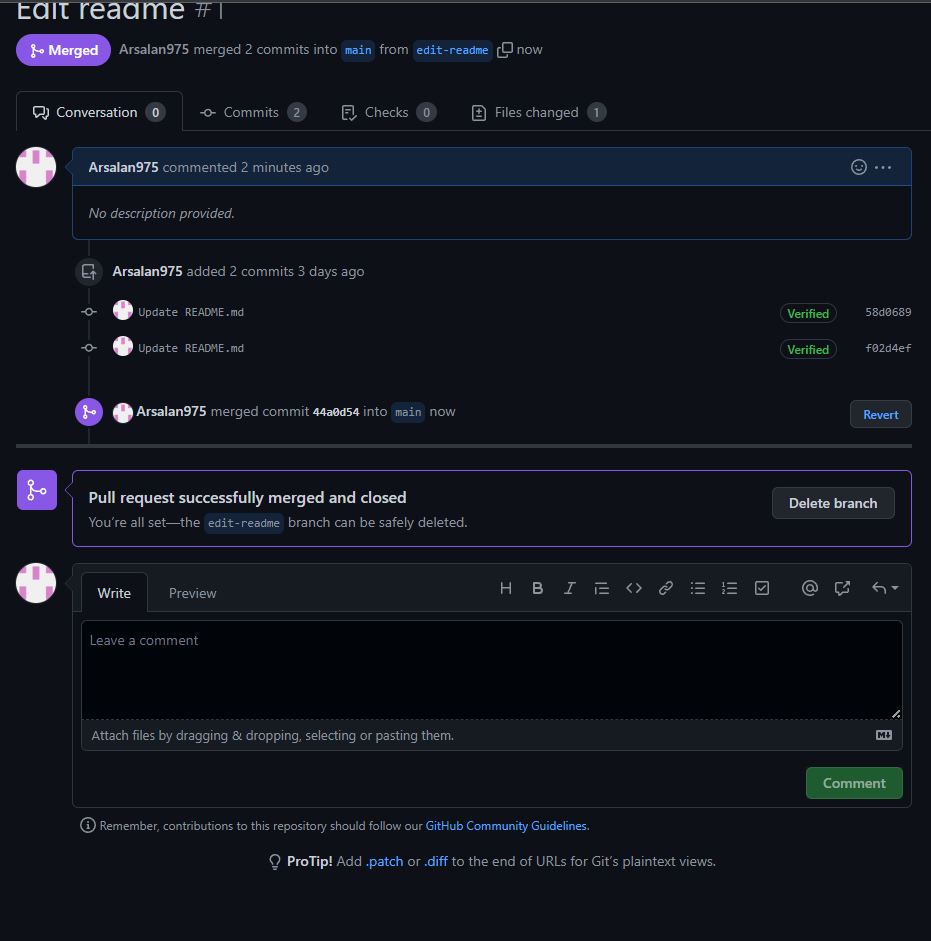
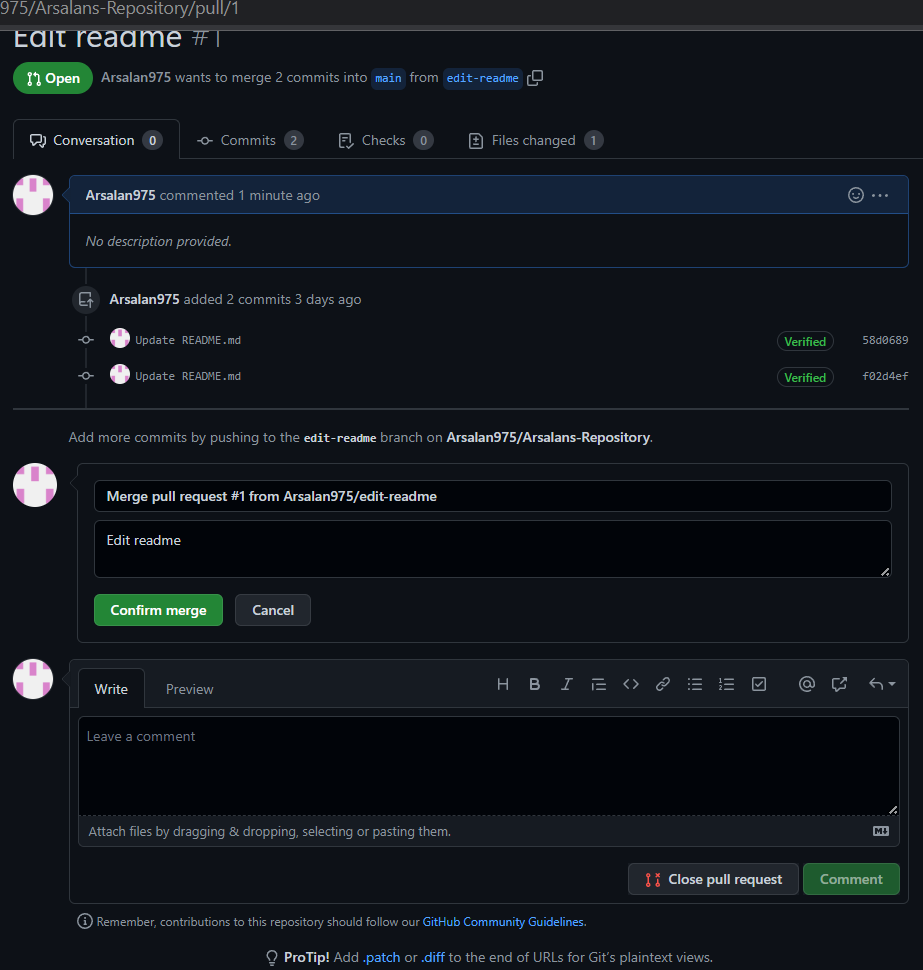
**Implement the following on GitHub:**  
**Assignment 1:**

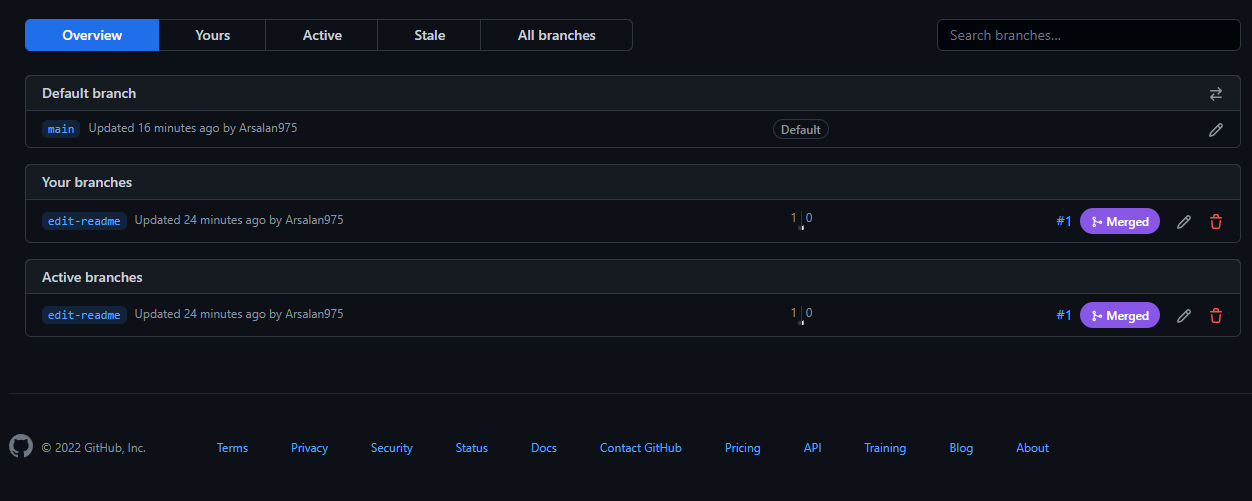
1. **Create an account on github.com.**

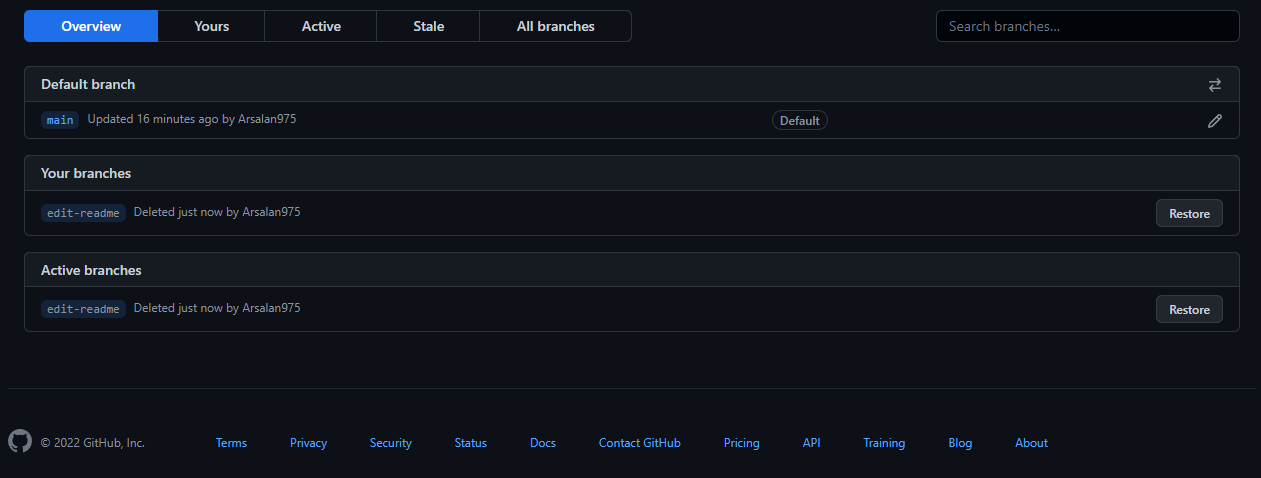
****

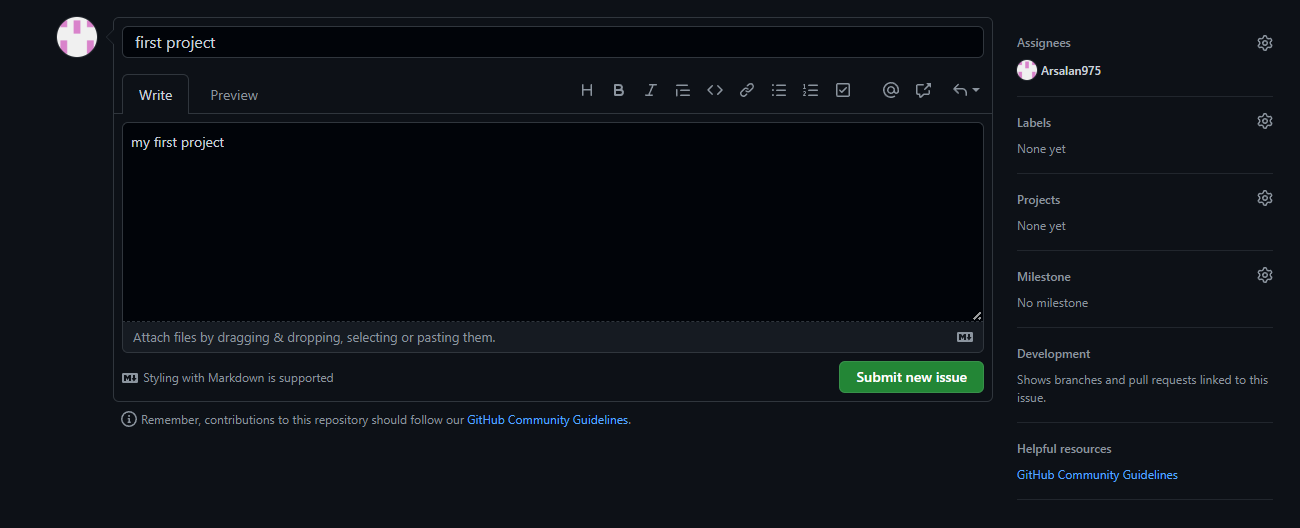
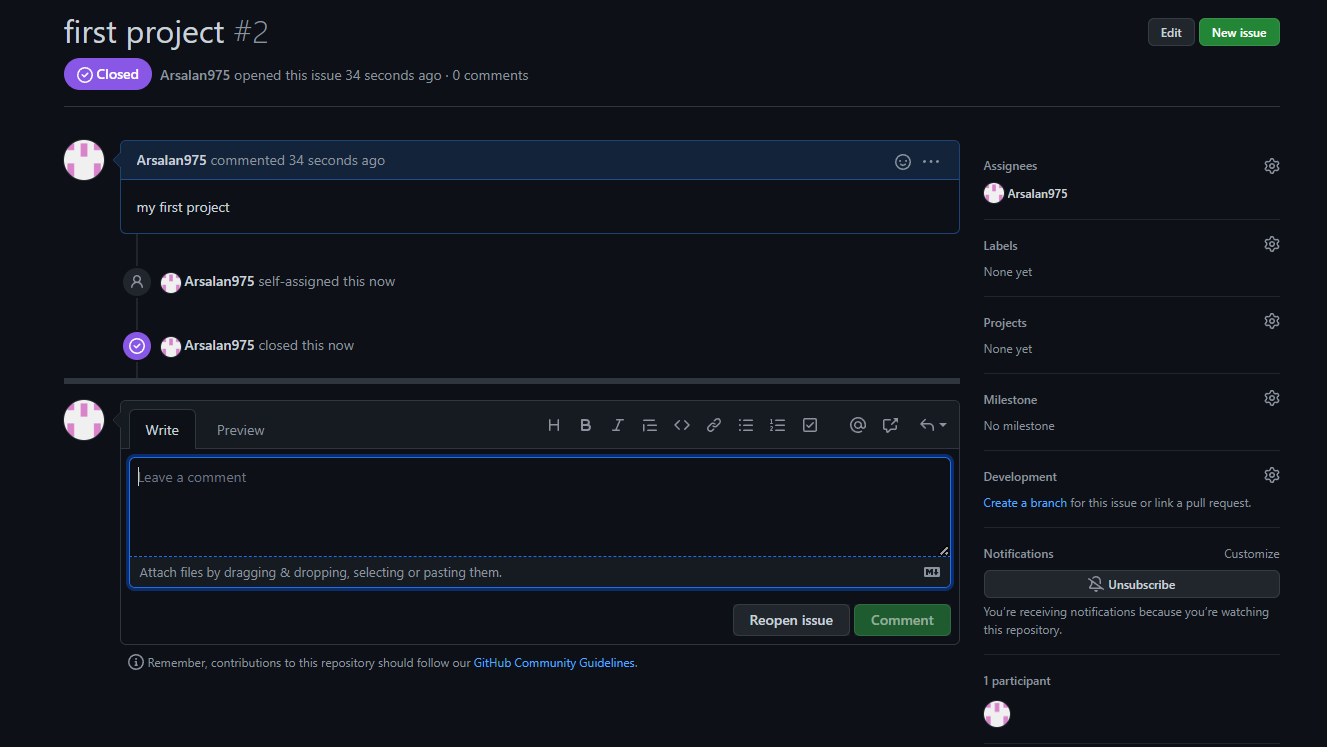
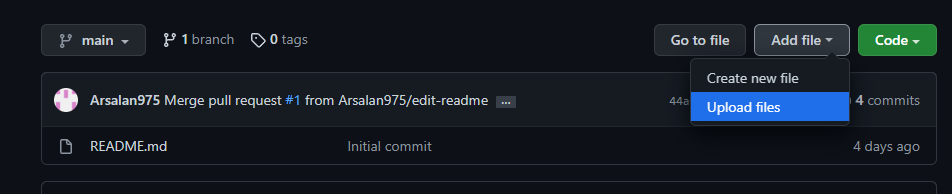
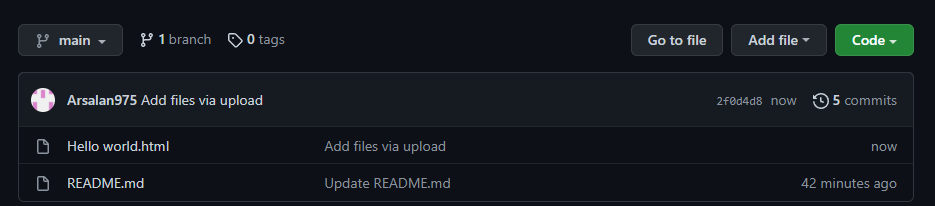
1. **Create a new Git repository.**
2. **Add the README.md file while creating the repository.**
3. **Create a new branch edit-readme.**
4. **Make changes to the README.md file with some markdown formatting from the cheat sheet given below.**
5. **Create a new pull request**.
6. **Compare and merge the changes to the main branch**

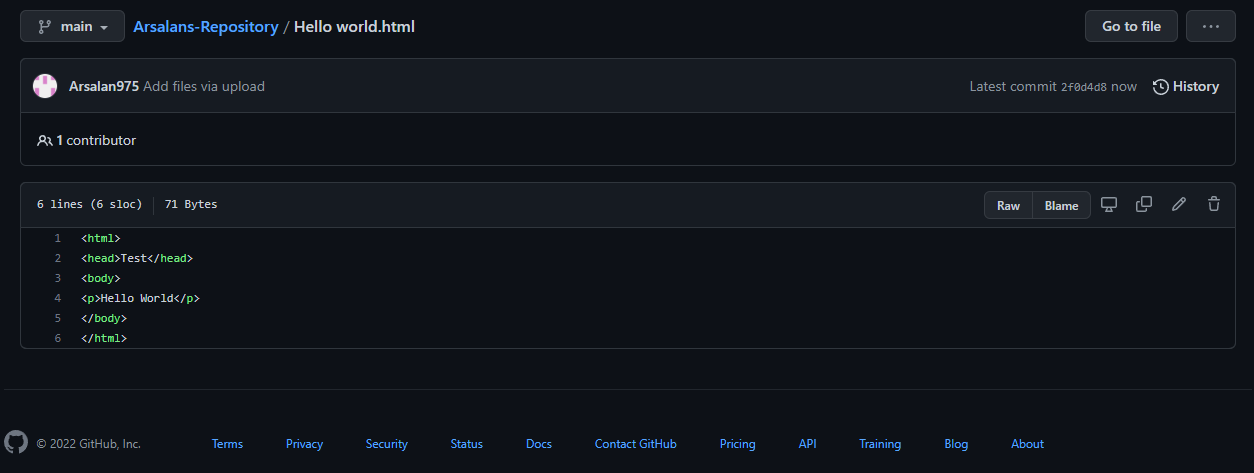
****

****

1. **Delete the edit-readme branch.**

****

1. **Create a new issue and assign it to yourself/ any other contributor.**
2. **Preview the issue and close it**
3. **Upload a file from your local drive to your Git repository.**



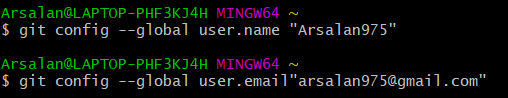
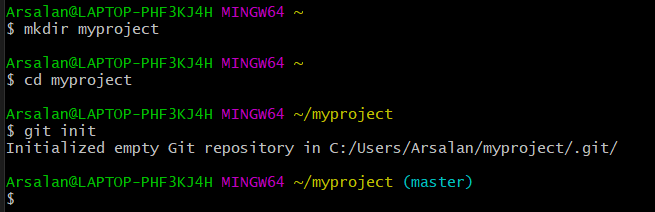
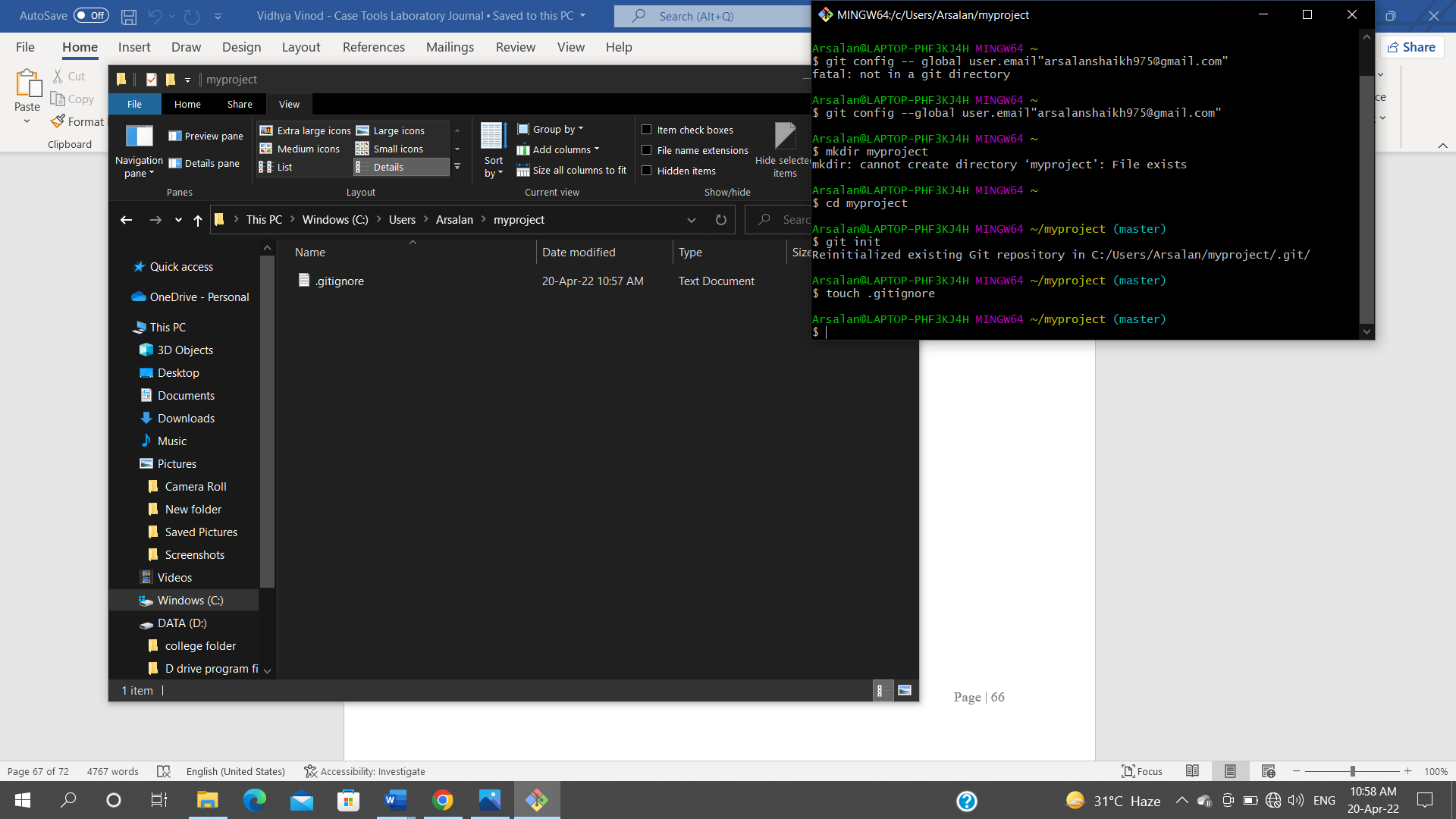
**SESSION 5: GIT \_\_\_\_\_\_\_\_**

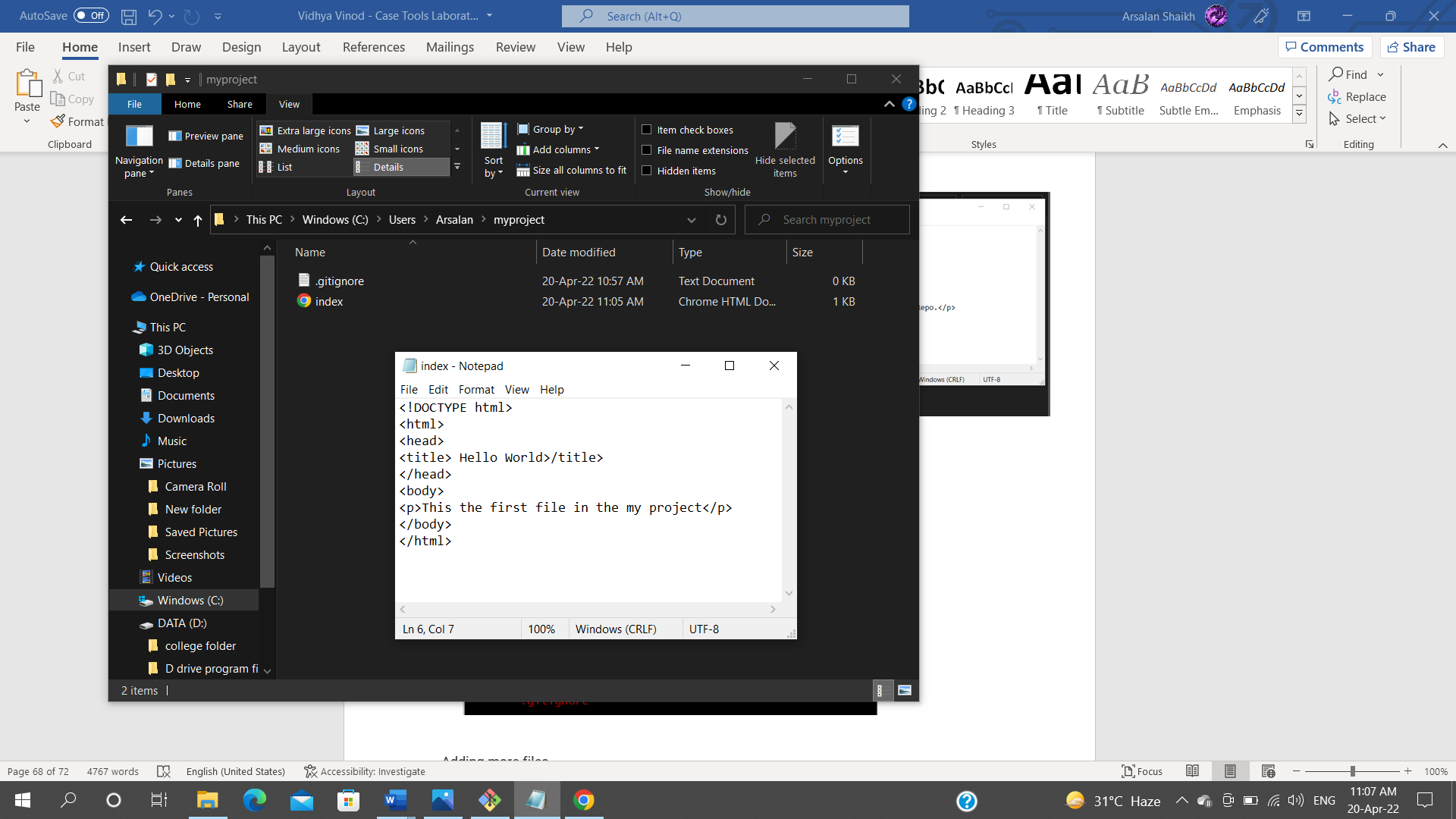
**Objective:** Learn to use centralised repositories and versioning tool.

**Date**: 30-03-2022/31-03-2022

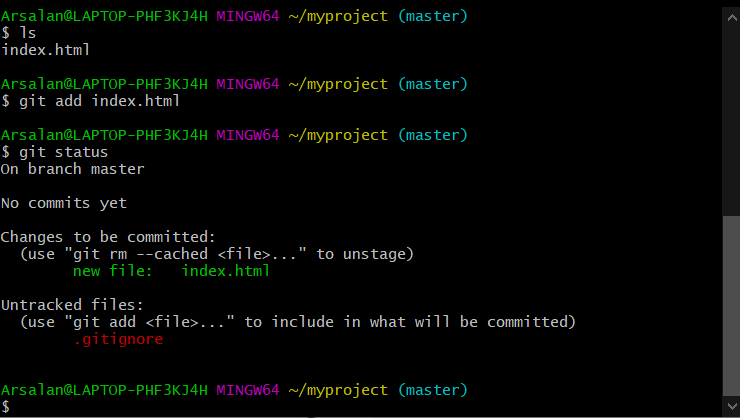
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

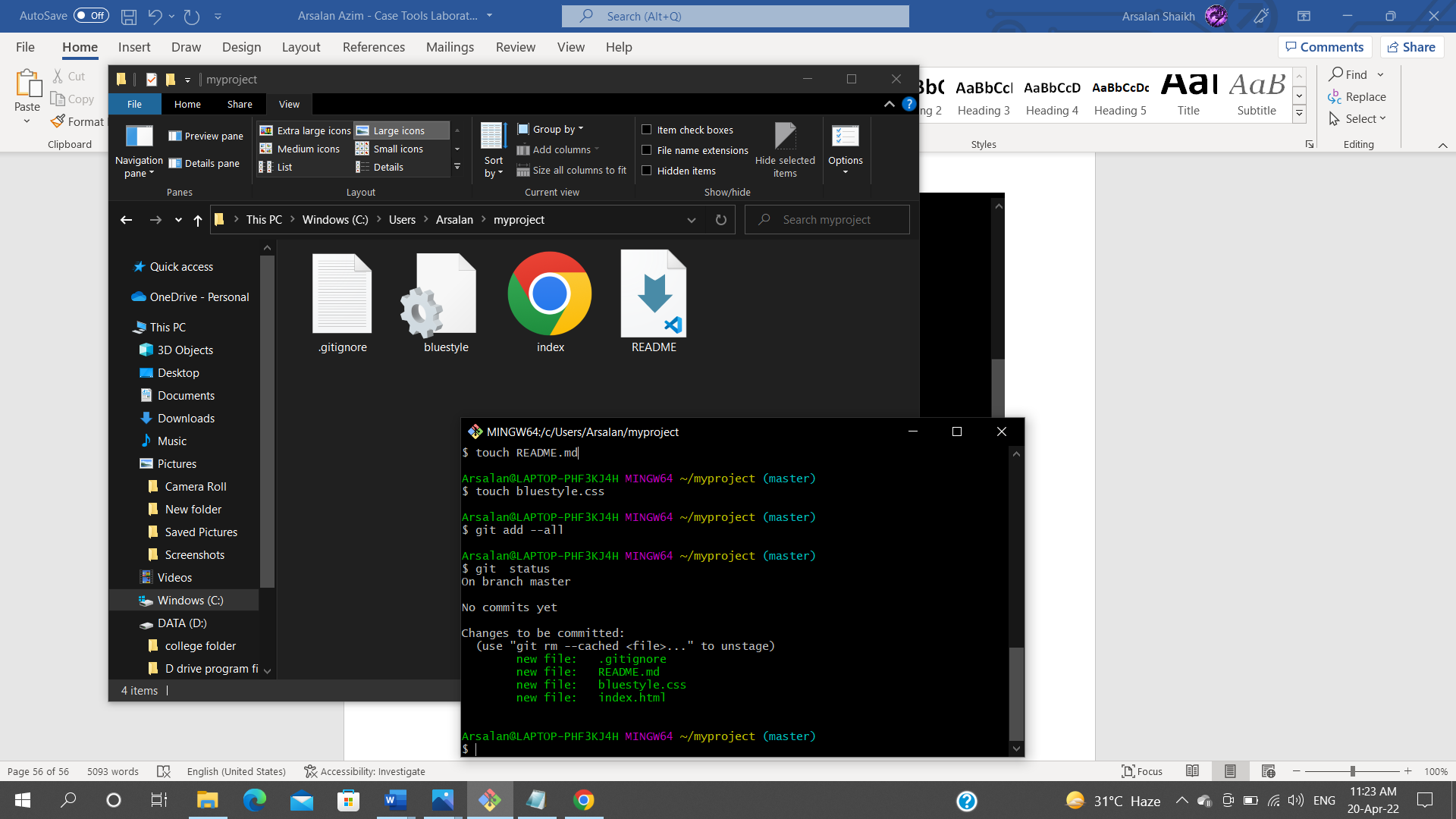
**Assignment 2:**

1. Configure Git Bash tool with the account credentials you created in the previous session.
2. Create a new local Git repository in your project working directory.
3. Create a .gitignore file and add the files not to be tracked.

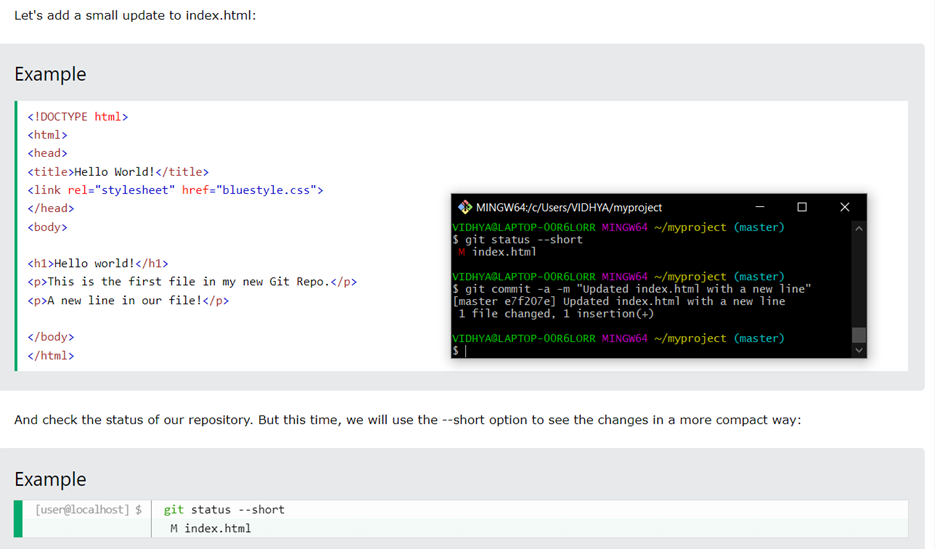
**4. Add files to the repository and perform an initial commit**

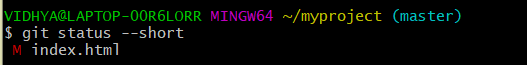
Stagging environment

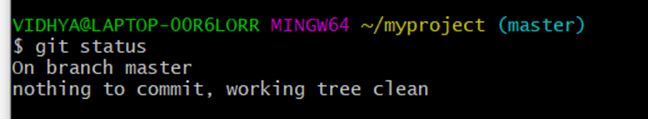


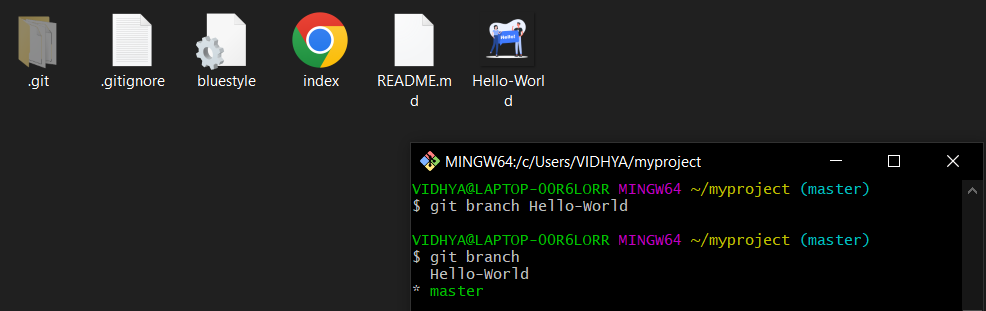
Adding more files

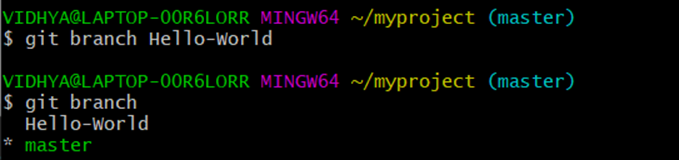
Commit

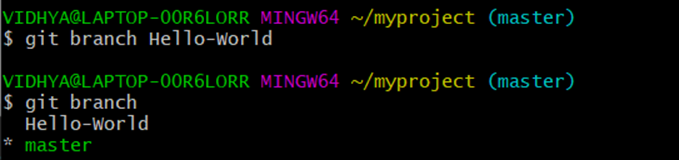
5. Modify a file and perform a second commit.

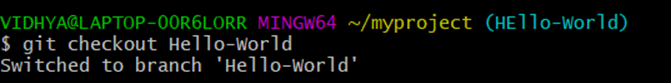
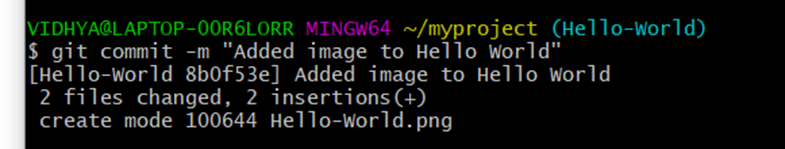
1. Display the repository status

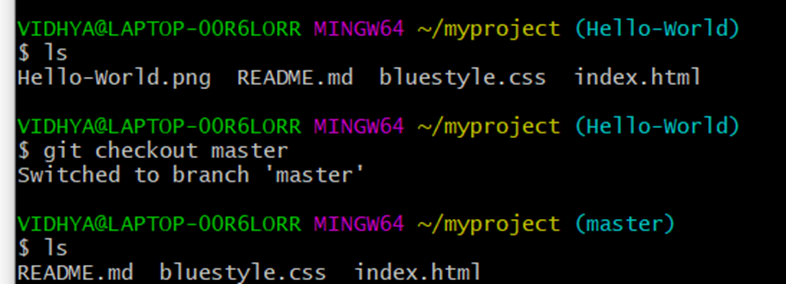


1. Create a new branch.
2. Edit some files in the branch and merge the code with the master branch.









1. Add the repository to github.com.

$ git remote add origin https://github.com/arsalan975/practise.git

$ git push --set-upstream origin master

1. Learn to clone a repository.

$ git clone https://github.com/arsalan975/practise.git