CAC-115: CASE TOOLS LABORATORY

Practical Journal

Name: Hemant Mhalsekar

Roll No: 21

Class: S.Y.B.C.A

Division: ’A’

Semester: IV

Year: 2021-2022

Dnyanprassarak Mandal’s

College and Research Centre, Assagao- Goa.

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Session No.** | **Title** | **Date** | **Page No.** | **Teacher’s Signature** |
| 1. | Code Documentation Tool- Javadoc | 09/03/2022 | 2 |  |
| 2. | Pair Programming | 23/03/2022 | 24 |  |
| 3. | Git (Assignment 1) | 06/04/2022 | 35 |  |
| 4. | Git (Assignment 2) | 12/04/2022 | 40 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**SESSION 1: Code Documentation using Javadoc**

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

**Date**: 10/03/2022

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

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

/\*\*

\* This program calculates the average marks of a student

\* **@author** Hemant

\*

\*/

**public** **class** AverageMain {

**float** english;

**float** maths;

**float** science;

/\*\*

\* This method is used to return the values for the variable english

\* **@return** english

\*/

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

**return** english;

}

/\*\*

\* This method is used to accept the values for the variable english

\* **@param** english

\*/

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

**this**.english = english;

}

/\*\*

\* This method is used to return the values for the variable maths

\* **@return** maths

\*/

**public** **float** getMaths() {

**return** maths;

}

/\*\*

\* This method is used to accept the values for the variable maths

\* **@param** maths

\*/

**public** **void** setMaths(**float** maths) {

**this**.maths = maths;

}

/\*\*

\* This method is used to return the values for the variable science

\* **@return** science

\*/

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

**return** science;

}

/\*\*

\* This method is used to accept the values for the variable science

\* **@param** science

\*/

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

**this**.science = science;

}

}

/\*\*

\* this is the main program

\* @author Hemant

\*

\*/

public class studentavg {

/\*\*

\* @param args this is the main class

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

avgmks a = new avgmks();

a.setEnglish(40);

a.setMaths(35);

a.setScience(29);

int avg = ( a.getEnglish()+a.getMaths()+a.getScience())/3;

System.out.println(" the average is "+ avg);

}

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

import java.util.Scanner;

public class younger {

/\*\*

\* this is the main function

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

int num1 = 0;

int num2 = 0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the age of two persons");

num1 = sc.nextInt();

num2 = sc.nextInt();

if(num1<num2)

System.out.println("person 1 is younger");

else if(num2<num1)

System.out.println("person 2 is younger");

else if(num2==num1)

System.out.println("both are of same age");

else

System.out.println("invalid");

}

}

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

members.

/\*\*

\*

\* @author Hemant

\*

\*/

public class items {

int id;

String name;

/\*\*

\* function used to return the id

\* @return the id

\*/

public int getId() {

return id;

}

/\*\*

\* function used to accept valus for id

\* @param id

\*/

public void setId(int id) {

this.id = id;

}

/\*\*

\* function used to return the name

\* @return the name

\*/

public String getName() {

return name;

}

/\*\*

\* function used to accept valus for name

\* @param name

\*/

public void setName(String name) {

this.name = name;

}

}

public class itemmain {

/\*\*

\* @author nelia

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

items i = new items();

i.setId(73);

i.setName("hemant");

System.out.println(" id "+ i.getId());

System.out.println(" name "+ i.getName());

}

4. 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.

}/\*\*

\*

\* @author Hemant

\*

\*/

public class stud

{

int rollNo;

int age;

String name;

/\*\*

\* function used to return the rollNo

\* @return the rollNo

\*/

public int getRollNo() {

return rollNo;

}

/\*\*

\* @param rollNo

\*/

public void setRollNo(int rollNo) {

this.rollNo = rollNo;

}

/\*\*

\* function used to return the age

\* @return the age

\*/

public int getAge() {

return age;

}

/\*\*

\* @param age

\*/

public void setAge(int age) {

this.age = age;

}

/\*\*

\* function used to return the name

\* @return the name

\*/

public String getName() {

return name;

}

/\*\*

\* @param name

\*/

public void setName(String name) {

this.name = name;

}

}

5. 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.

/\*\*

\* This program displays the details of the cars

\* @author Hemant

\*

\*/

public class Car {

String name;

String color;

Page | 12

int cost;

/\*\*

\* This method returns the name of the car

\* @return name

\*/

public String getName() {

return name;

}

/\*\*

\* This method sets the name of the car

\* @param name

\*/

public void setName(String name) {

this.name = name;

}

/\*\*

\* This method returns the color of the car

\* @return color

\*/

public String getColor() {

return color;

}

/\*\*

\* This method sets the color for the car

\* @param color

\*/

public void setColor(String color) {

this.color = color;

}

/\*\*

\* This method returns the cost of the car

\* @return cost

\*/

public int getCost() {

return cost;

}

/\*\*

\* This method sets the cost

\* @param cost

\*/

public void setCost(int cost) {

this.cost = cost;

}

}

public class CarMain {

public static void main(String[] args) {

// TODO Auto-generated method stub

Car C1 = new Car();//Object is created for car 1

Car C2 = new Car();//Object is create for car 2

C1.setName("Renault Duster");

C1.setColor("Honey-Brown");

C1.setCost(1100000);

C2.setName("Volkswagen Polo");

C2.setColor("White");

C2.setCost(1000000);

System.out.println("Car 1");

System.out.println("Car name--- "+C1.getName());

System.out.println("Color--- "+C1.getColor());

System.out.println("Cost---"+C1.getCost());

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_");

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

System.out.println("Car name--- "+C2.getName());

System.out.println("Color--- "+C2.getColor());

System.out.println("Cost---"+C2.getCost());

}

}

Output:

Car 1

Car name--- Renault Duster

Color--- Honey-Brown

Cost---1100000

\_\_\_\_\_\_\_\_\_\_\_\_

Car 2

Car name--- Volkswagen Polo

Color--- White

Cost---1000000

6. 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]

/\*\*

\* This program calculates the areas of different shapes.

\* @author Hemant

\*

\*/

public class Shape {

double radius;

int length,breadth;

int side;

int base,height;

/\*\*

\* This method sets the radius of the circle and calculates the area of the circle.

\* @param r

\*/

void CircleArea(double r)

{

radius=r;

double A = 3.14\*r\*r;

System.out.println("The area of circle---"+A);

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

System.out.println("\n");

}

/\*\*

\* This method sets the parameters for length and breadth of the rectangle and calculates the

area of rectangle

\* @param x

\* @param y

\*/

void RectangleArea(int x, int y)

{

length=x;

breadth=y;

int A = length\*breadth;

System.out.println("The area of rectangle----"+A);

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

System.out.println("\n");

}

/\*\*

\* This method set the parameters for side of a square and calculates the area of square

\* @param x

\*/

void SquareArea(int x)

{

side=x;

int A = side\*side;

System.out.println("The area of square is----"+A);

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

System.out.println("\n");

}

/\*\*

\* This method sets the method for base and height of the triangle and calculates the area of

the triangle.

\* @param x

\* @param y

\*/

void TriangleArea(int x, int y)

{

base=x;

height=y;

double A= 0.5\*base\*height;

System.out.println("The area of triangle is----"+A);

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

}

}

public class ShapeMain {

public static void main(String[] args) {

// TODO Auto-generated method stub

Shape S1 = new Shape();

S1.CircleArea(4);

S1.RectangleArea(2, 5);

S1.SquareArea(6);

S1.TriangleArea(3, 7);

}

}

Output:

The area of circle---50.24

\*\*\*\*\*\*\*\*\*

The area of rectangle----10

\*\*\*\*\*\*\*\*\*

The area of square is----36

\*\*\*\*\*\*\*\*\*

The area of triangle is----10.5

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

/\*\*

\* This program displays the Product details using constructor

\* @author Hemant

\*

\*/

public class Product {

int id;

String name;

int cost;

/\*\*

\* Non-parameterized Constructor

\*/

public Product()

{

Page | 18

id=10023;

name="Nivea Body Lotion";

cost=250;

}

/\*\*

\* Parameterized constructor

\* @param id

\* @param name

\* @param cost

\*/

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

super();

this.id = id;

this.name = name;

this.cost = cost;

}

/\*\*

\* This method displays the product details.

\*/

void display()

{

System.out.println("Product Id---"+id);

System.out.println("Product name---"+name);

System.out.println("Product Cost---"+cost);

System.out.println("\*\*\*\*\*\*\*\*\n");

}

}

public class ProductMain {

public static void main(String[] args) {

// TODO Auto-generated method stub

Product P1 = new Product();

Product P2 = new Product(20031,"Lux Body Wash",99);

P1.display();

P2.display();

}

}

Output:

Product Id---10023

Product name---Nivea Body Lotion

Product Cost---250

\*\*\*\*\*\*\*\*

Product Id---20031

Product name---Lux Body Wash

Product Cost---99

8. 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.

import java.util.Scanner;

/\*\*

\* This program displays and calculates final amount of Vehicle

\* @author Hemant

\*

\*/

public class Vehicle {

String name, brand;

int cost, roadTax,op;

Sanner sc = new Scanner(System.in);

/\*\*

\* This method returns name

\* @return name

\*/

public String getName() {

return name;

}

/\*\*

\* This method sets the name of the vehicle

\* @param name

\*/

public void setName(String name) {

this.name = name;

}

/\*\*

\* This method returns Brand

\* @return brand

\*/

public String getBrand() {

return brand;

}

/\*\*

\* This method sets the Brand of the vehicle.

\* @param brand

\*/

public void setBrand(String brand) {

this.brand = brand;

}

/\*\*

\* This method returns the cost

\* @return cost

\*/

public int getCost() {

return cost;

}

/\*\*

\* This method sets the cost of the vehicle.

\* @param cost

\*/

public void setCost(int cost) {

this.cost = cost;

}

/\*\*

\* This method is for calculating the Road Tax.

\*/

void calculateRoadTax()

{

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

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

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

System.out.println("Enter your option---");

int op=sc.nextInt();

if(op==1)

{

roadTax=500;

}

if(op==2)

{

roadTax=1000;

}

if(op==3)

{

roadTax=1500;

}

}

}

public interface Amount {

int govtTax=2000;

/\*\*

\* This method is for calculating the government Tax

\*/

void calc();

}

public class Result extends Vehicle implements Amount {

/\*\*

\* This method is to calculate the final amount of the vehicle

\*/

@Override

public void calc() {

// TODO Auto-generated method stub

int finalAmount=cost+roadTax+govtTax;

System.out.println("Your Final Amount----"+finalAmount);

}

}

import java.util.Scanner;

public class ResultMain {

public static void main(String[] args) {

// TODO Auto-generated method stub

int cost;

String name, brand;

Result R1 = new Result();

Scanner sc = new Scanner(System.in);

R1.getName();

System.out.println("Enter your vehicle's name---");

name=sc.nextLine();

name=sc.nextLine();

R1.setName(name);

R1.getBrand();

System.out.println("Enter your vehicle's Brand---");

brand=sc.nextLine();

brand=sc.nextLine();

R1.setBrand(brand);

R1.getCost();

System.out.println("Enter your vehicle's cost---");

cost=sc.nextInt();

R1.setCost(cost);

R1.calculateRoadTax();

R1.calc();

}

}

Output:

Enter your vehicle's name---

Renault Duster

Enter your vehicle's Brand---

Renault

Enter your vehicle's cost---

1000000

1. Bike

2. Car

3. Truck

Enter your option---

2

Your Final Amount----1003000

9. Demonstrate divide by zero Exception.

/\*\*

\* This program displays Arithimetic Exception.

\* @author Hemant

\*/

public class Divide {

public static void main(String[] args) {

// TODO Auto-generated method stub

int x=0,y=0,d=10,b=5,c=5;

try

{

x=d/(b-c);

}

catch(ArithmeticException e)

{

System.out.println("Arithmetic exception");

}

}

}

Output:

Error!!

Arithmetic exception

10. Demonstrate ArrayOutOfBounds Exception.

/\*\*

\* This method display Array Index Out Of Bounds Exception.

\* @author Hemant

\*/

public class Demo {

public static void main(String[] args) {

// TODO Auto-generated method stub

int a[]= new int[5];

try

{

a[6]=8;

}

catch(ArrayIndexOutOfBoundsException e)

Page | 26

{

System.out.println("Error!!");

System.out.println("Array out of Bounds Exception");

}

}

}

Output:

Error!!

Array out of Bounds Exception

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

**Session 3:** Pair Programming

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

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

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

1.

/\*\*

\* This program gets the area of the circle

\* **@author** Hemant and Nelia

\*

\*/

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

**private** **double** radius =1.0;

**private** String colour="Red";

/\*\*

\* This is a non parameterised constructor Circle

\*/

**public** Circle() {

radius = 2.0;

colour = "Blue";

}

/\*\*

\* This is a parameterised constructor Circle

\* **@param** r

\*/

**public** Circle(**double** r) {

radius = r;

}

/\*\*

\* **@return** the radius

\*/

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

**return** radius;

}

/\*\*

\* This method returns Area of Circle

\* **@return** Area of the Circle

\*/

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

**return** 3.14\*radius\*radius;

}

}

/\*\*

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** CircleTest {

/\*\*

\* **@param** args main

\*/

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

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

Circle c1 = **new** Circle();

System.***out***.println("The Circle has a Radius of "+c1.getRadius()+ " and the Area is "+ c1.getArea());

Circle c2 = **new** Circle(3.0);

System.***out***.println("The Circle has a Radius of "+c2.getRadius()+ " and the Area is "+ c2.getArea());

}

}

Output:

The Circle has a Radius of 2.0 and the Area is 12.56

The Circle has a Radius of 3.0 and the Area is 28.259999999999998

2.

/\*\*

\* this program displays day,month,year,date

\* **@author** Hemant and nelia

\*

\*/

**public** **class** Date {

**int** day;

**int** month;

**int** year;

**int** date;

/\*\*

\* this method returns the day

\* **@return** the day

\*/

**public** **int** getDay() {

**return** day;

}

/\*\*

\* this method sets the day

\* **@param** day

\*/

**public** **void** setDay(**int** day) {

**this**.day = day;

}

/\*\*

\* this method returns the month

\* **@return** the month

\*/

**public** **int** getMonth() {

**return** month;

}

/\*\*

\* this method sets the month

\* **@param** month

\*/

**public** **void** setMonth(**int** month) {

**this**.month = month;

}

/\*\*

\* this method return the year

\* **@return** the year

\*/

**public** **int** getYear() {

**return** year;

}

/\*\*

\* this method sets the year

\* **@param** year

\*/

**public** **void** setYear(**int** year) {

**this**.year = year;

}

/\*\*

\* this method returns the date

\* **@return** the date

\*/

**public** **int** getDate() {

**return** date;

}

/\*\*

\* this method sets the date

\* **@param** date

\*/

**public** **void** setDate(**int** date) {

**this**.date = date;

}

**public** **void** printDate() {

System.***out***.println("mm/dd/yyyy = " + month + "/" + day + "/" + year);

}

}

/\*\*

\* this is the main method

\* **@author** Hemant and nelia

\*

\*/

**public** **class** DateTest {

/\*\*

\* **@param** args main

\*/

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

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

Date D1 = **new** Date();

D1.setDay(12);

D1.setMonth(02);

D1.setYear(2002);

System.***out***.println("Month "+D1.getMonth());

System.***out***.println("Day "+D1.getDay());

System.***out***.println("Year "+D1.getYear());

D1.printDate();

}

}

Output:

Month 2

Day 12

Year 2002

mm/dd/yyyy = 2/12/2002

3.

/\*\*

\* This program displays if the lights are on or not

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** Lamp {

**boolean** isOn;

/\*\*

\* Method to turn on the light

\*/

**void** turnOn() {

isOn = **true**;

System.***out***.println("Lights On? \n Yes!!!");

}

/\*\*

\* Method to turn off the light

\*/

**void** turnOff() {

isOn = **false**;

System.***out***.println("Lights on? \n No!!!");

}

}

/\*\*

\* This is the main Method

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** LampMain {

/\*\*

\* **@param** args main

\*/

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

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

Lamp led = **new** Lamp();

Lamp halogen = **new** Lamp();

// turn on the light by calling method turnOn()

led.turnOn();

// turn off the light by calling method turnOff()

halogen.turnOff();

}

}

Output:

Lights On?

Yes!!!

Lights on?

No!!!

4.

/\*\*

\* this program the details of the book

\* **@author** Hemant

\*

\*/

**public** **class** Book {

String author;

String title;

**int** ISBN;

/\*\*

\* this returns the name of the author

\* **@return** the author

\*/

**public** String getAuthor() {

**return** author;

}

/\*\*

\* this sets the name of the author

\* **@param** author

\*/

**public** **void** setAuthor(String author) {

**this**.author = author;

}

/\*\*

\* this returns the title of the book

\* **@return** the title

\*/

**public** String getTitle() {

**return** title;

}

/\*\*

\* this sets the title of the book

\* **@param** title

\*/

**public** **void** setTitle(String title) {

**this**.title = title;

}

/\*\*this returns the ISBN of the book

\* **@return** the iSBN

\*/

**public** **int** getISBN() {

**return** ISBN;

}

/\*\*

\* this sets ISBN of the book

\* **@param** iSBN

\*/

**public** **void** setISBN(**int** iSBN) {

ISBN = iSBN;

}

Book(String a,String b, **int** c)

{

author= a;

title= b;

ISBN= c;

}

/\*\*

\* Non-Parameterized Constructor

\*/

Book()

{

author= "Nelia";

title= "Big Dreams";

ISBN= 200046550;

}

/\*\*

\* This method displays output

\*/

**void** display()

{

System.***out***.println("The Book details\n Author:"+author+ "\n Title:"+title+ "\n ISBN:"+ISBN);

}

}

/\*\*

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** BookMain {

/\*\*

\* **@param** args main

\*/

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

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

Book B1= **new** Book();

B1.getAuthor();

B1.getTitle();

B1.getISBN();

B1.display();

B1.setAuthor("Korth");

B1.setTitle("DBMS");

B1.setISBN(2245678);

B1.display();

}

}

Output:

The Book details

Author:Nelia

Title:Big Dreams

ISBN:200046550

The Book details

Author:Korth

Title:DBMS

ISBN:2245678

5

/\*\*

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** Animal {

String name;

**public** **void** eat()

{

System.***out***.println("I can eat");

}

}

/\*\*

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** Dog **extends** Animal{

**public** **void** display() {

System.***out***.println("My name is " + name);

}

}

/\*\*

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** AnimalMain {

/\*\*

\* **@param** args main

\*/

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

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

Dog labrador = **new** Dog();

labrador.name = "Juno";

labrador.display();

labrador.eat();

}

}

Output:

My name is Juno

I can eat

6.

/\*\*

\* **@author** Hemant Nelia

\*

\*/

**public** **class** Employee {

**float** salary;

/\*\*

\* This method returns salary

\* **@return** the salary

\*/

**public** **float** getSalary() {

**return** salary;

}

/\*\*

\* This method sets the salary

\* **@param** salary

\*/

**public** **void** setSalary(**float** salary) {

**this**.salary = salary;

}

**public** **void** Pdisplay() {

System.***out***.println("The salary of the employee is "+getSalary());

}

}

/\*\*

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** Programmer **extends** Employee {

/\*\*

\* **@param** args main

\*/

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

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

**int** bonus=1500;

Employee E1 = **new** Employee();

E1.setSalary(50000);

E1.Pdisplay();

}

}

Output:

The salary of the employee is 50000.0

7.

/\*\*

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** BankAcc {

**double** balance;

**public** BankAcc(**double** balance) {

**this**.balance = balance;

}

**public** **void** deposit(**double** amount) {

**this**.balance += amount;

}

**public** **void** withraw(**double** amount) {

**this**.balance -= amount;

}

**public** **double** getBalance() {

**return** balance;

}

}

/\*\*

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** SavingsAcc **extends** BankAcc{

**float** interest = 0.01F;

/\*\*

\* **@param** balance

\*/

**public** SavingsAcc(**double** balance) {

**super**(balance);

// **TODO** Auto-generated constructor stub

}

**public** **void** addInterest() {

balance += balance \* interest;

}

}

/\*\*

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** CheckingAcc **extends** BankAcc{

**float** deductAmount = 0.0025F;

/\*\*

\* **@param** balance

\*/

**public** CheckingAcc(**double** balance) {

**super**(balance);

// **TODO** Auto-generated constructor stub

}

**public** **void** deductFees() {

balance -= balance \* deductAmount;

}

}

/\*\*

\* **@author** Hemant and Nelia

\*

\*/

**public** **class** AccMain {

/\*\*

\* **@param** args

\*/

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

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

SavingsAcc sa = **new** SavingsAcc(50000);

CheckingAcc ca = **new** CheckingAcc(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

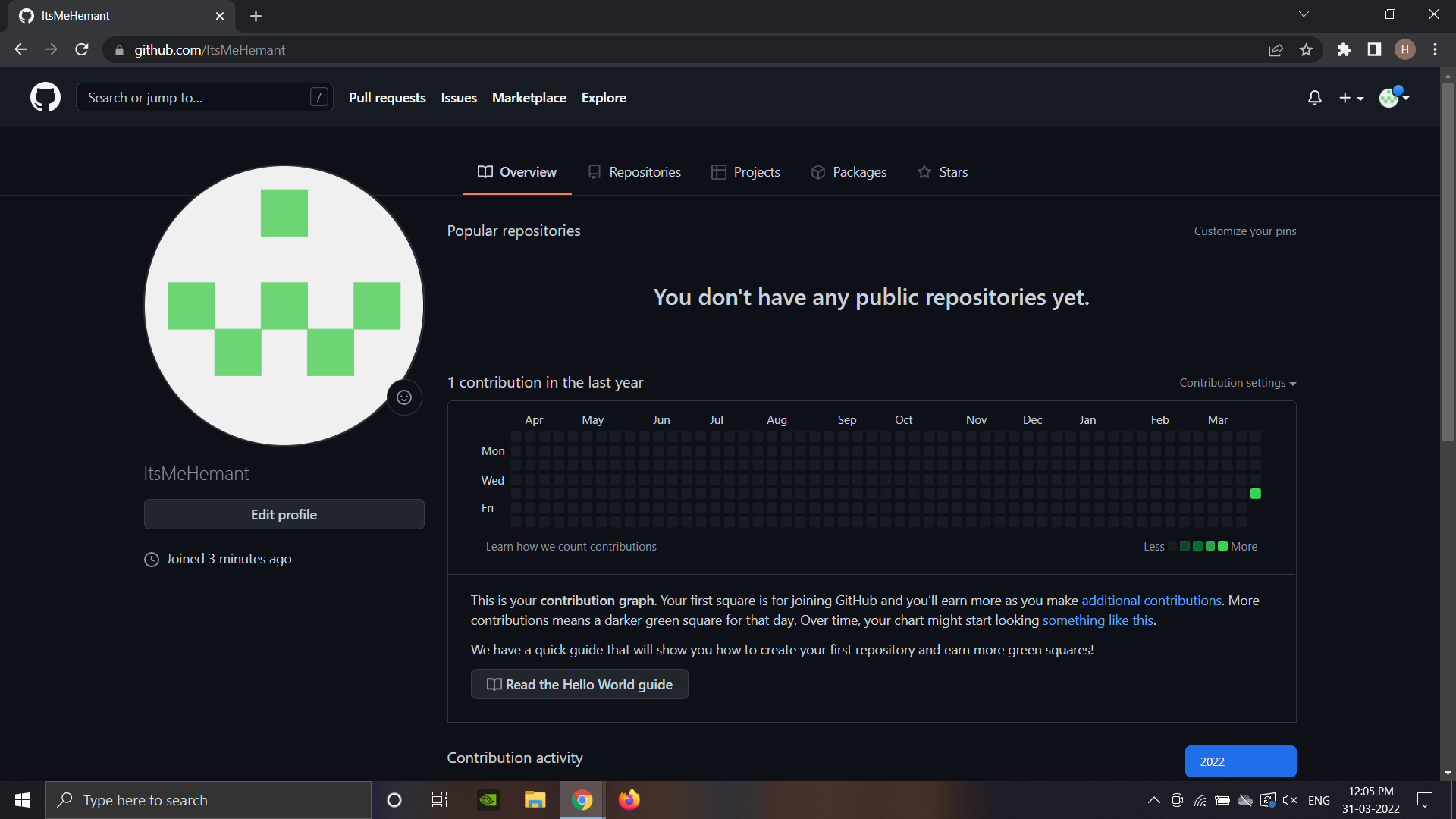
After adding and deducting

CA = 997500.0000558794

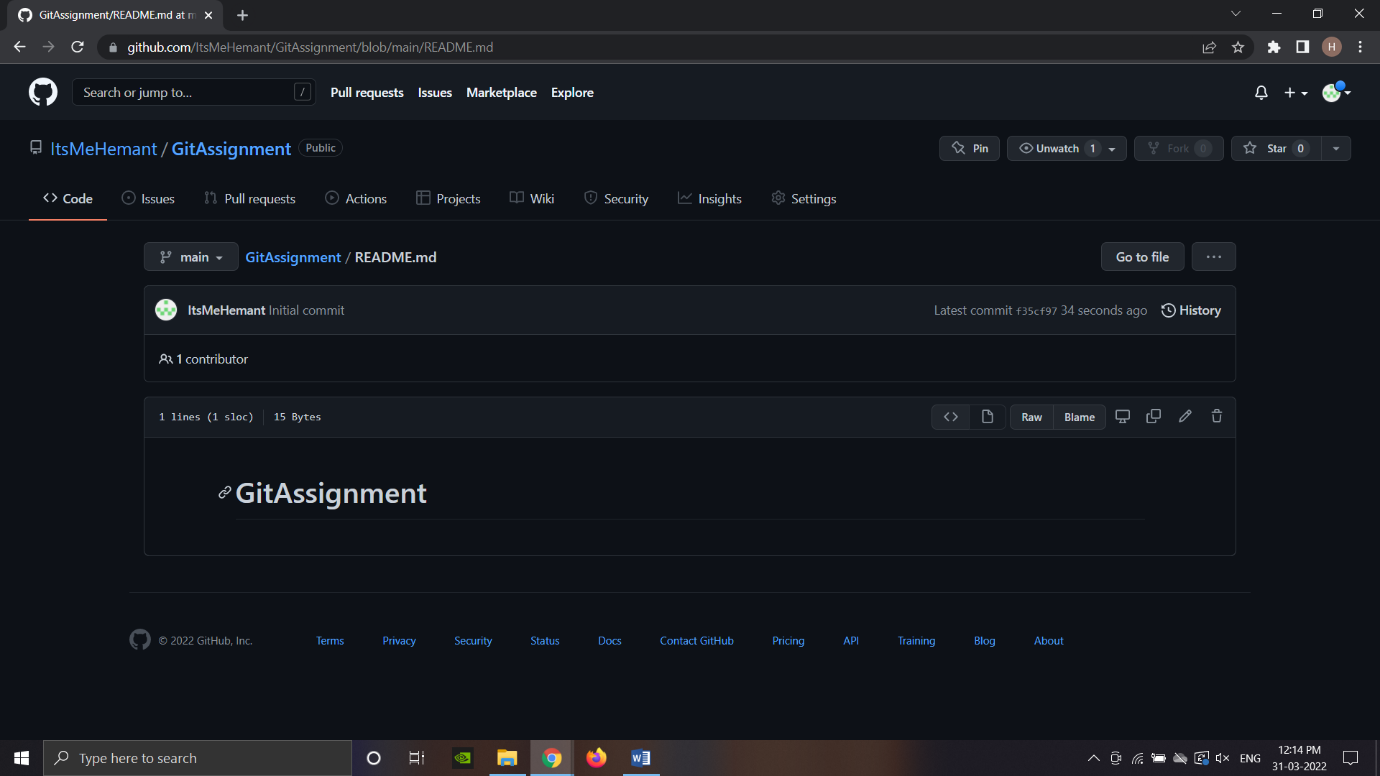
SA = 50499.99998882413

**Session: GIT  
Objective: Learn to use centralized repositories and versioning tool.  
Date: 30-03-2022/31-03-2022  
Implement the following on GitHub:**

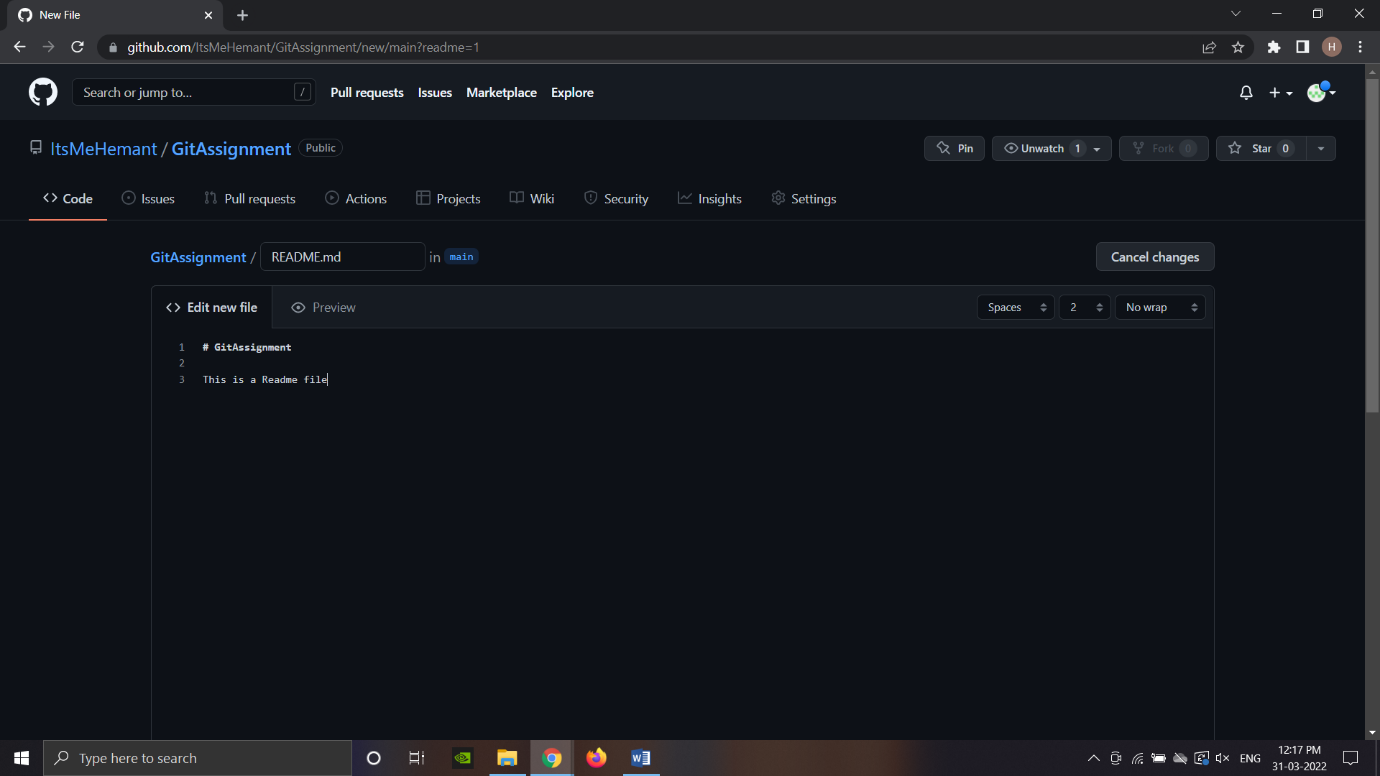
**1. Create an account on github.com.**



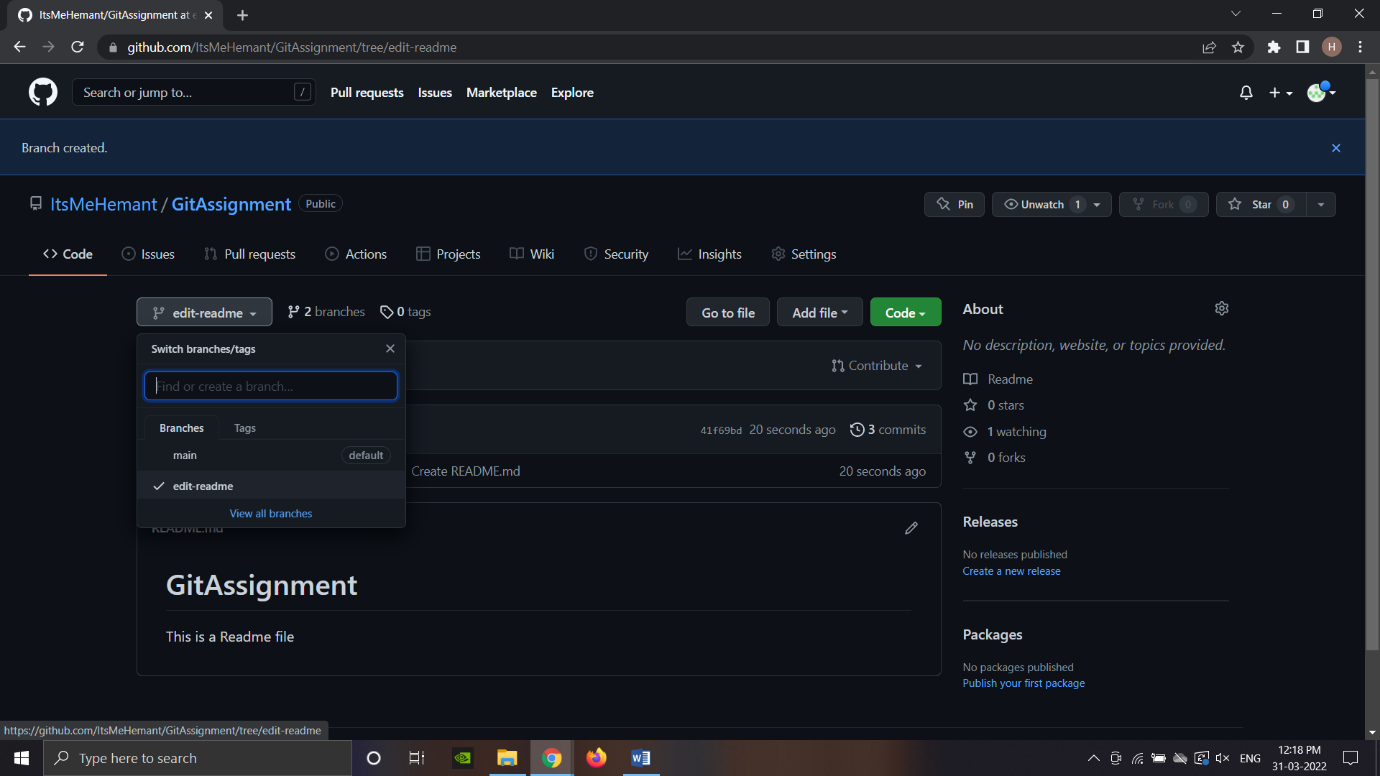
**2. Create a new Git repository.**



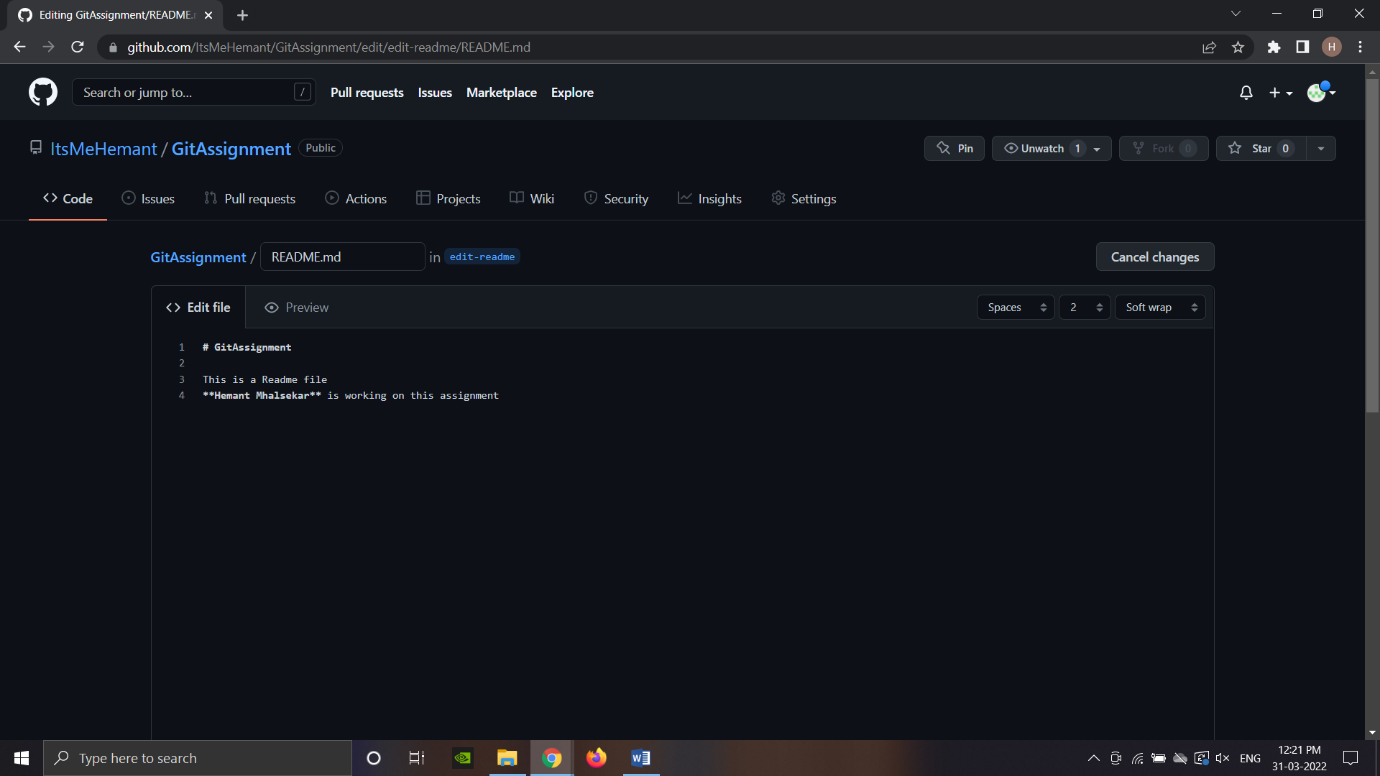
**3. Add the README.md file while creating the repository.**



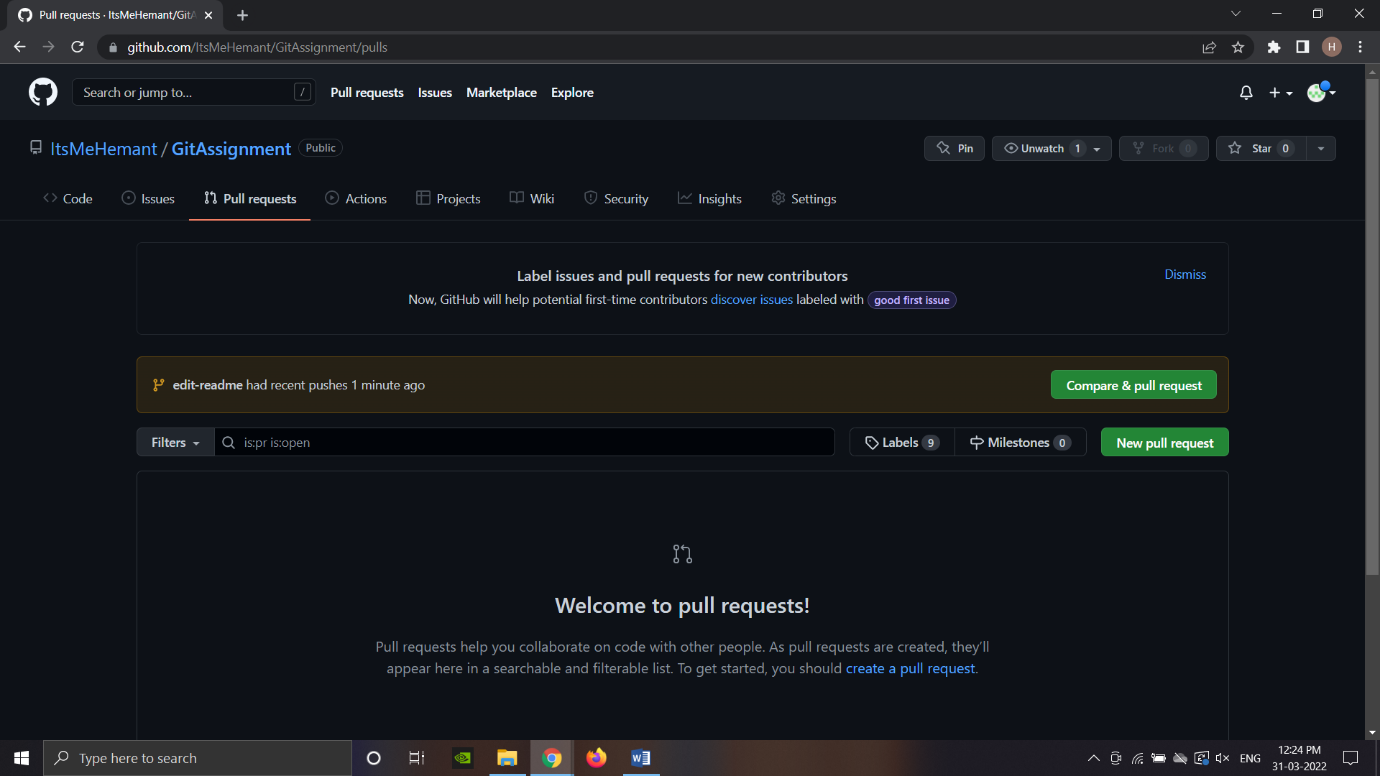
**4. Create a new branch edit-readme.**



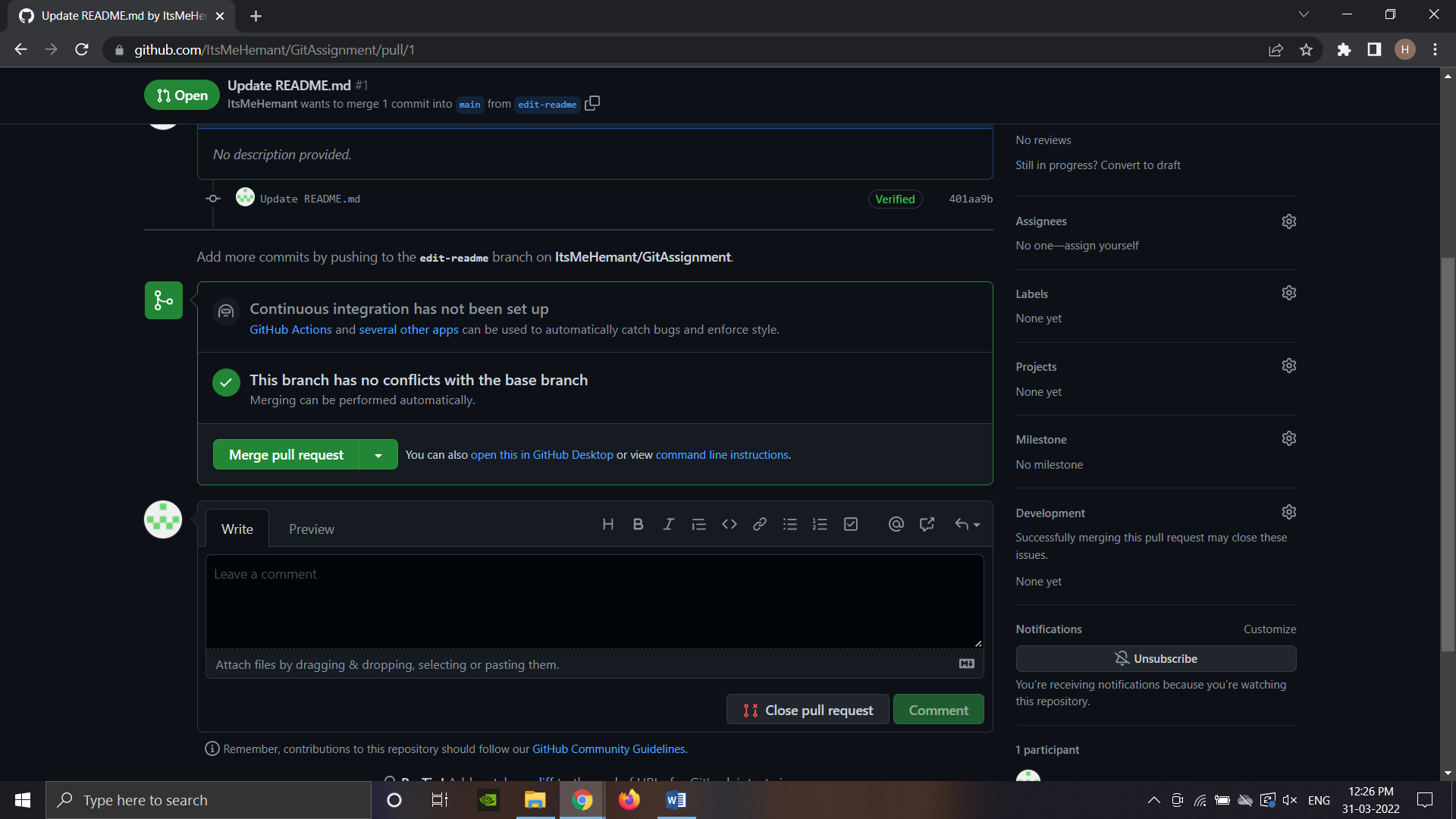
**5. Make changes to the README.md file with some markdown formatting from the cheat sheet given below.**



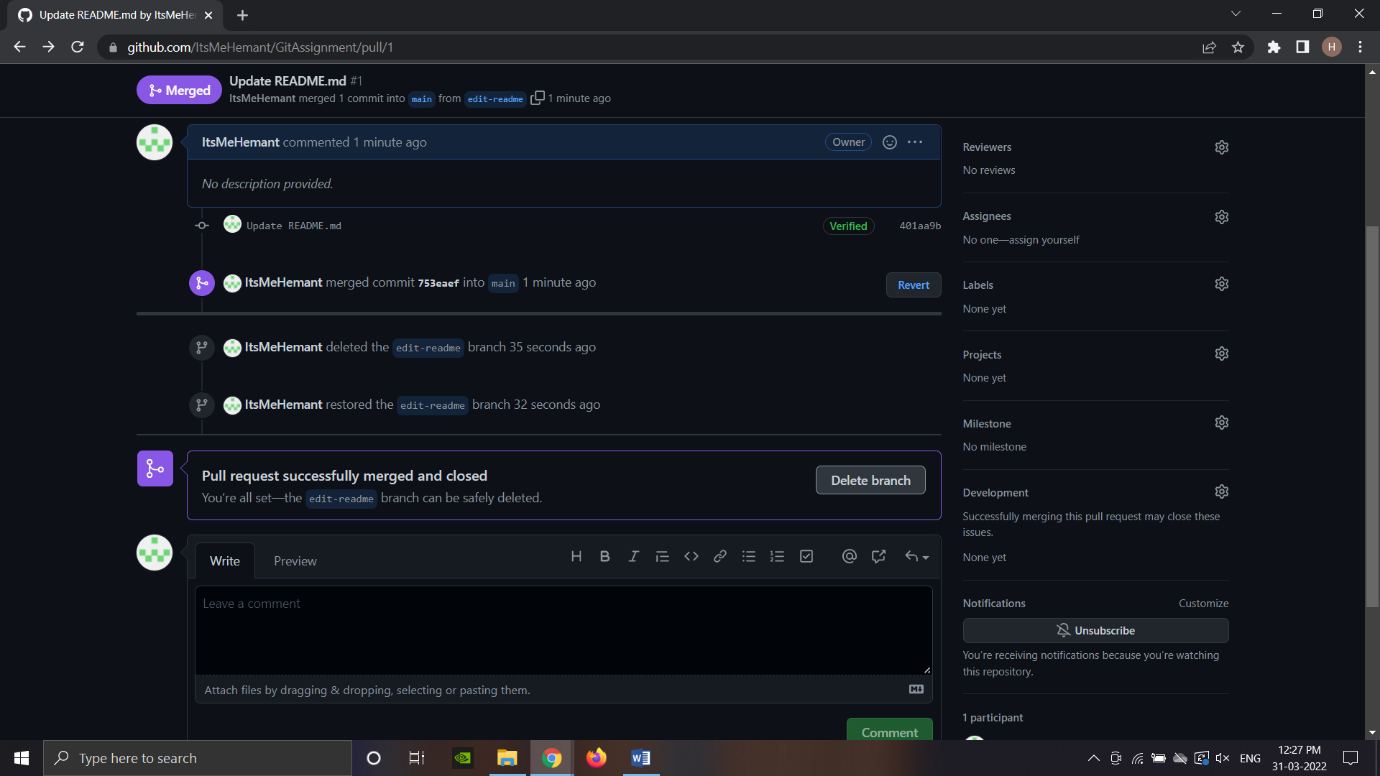
**6. Create a new pull request.**



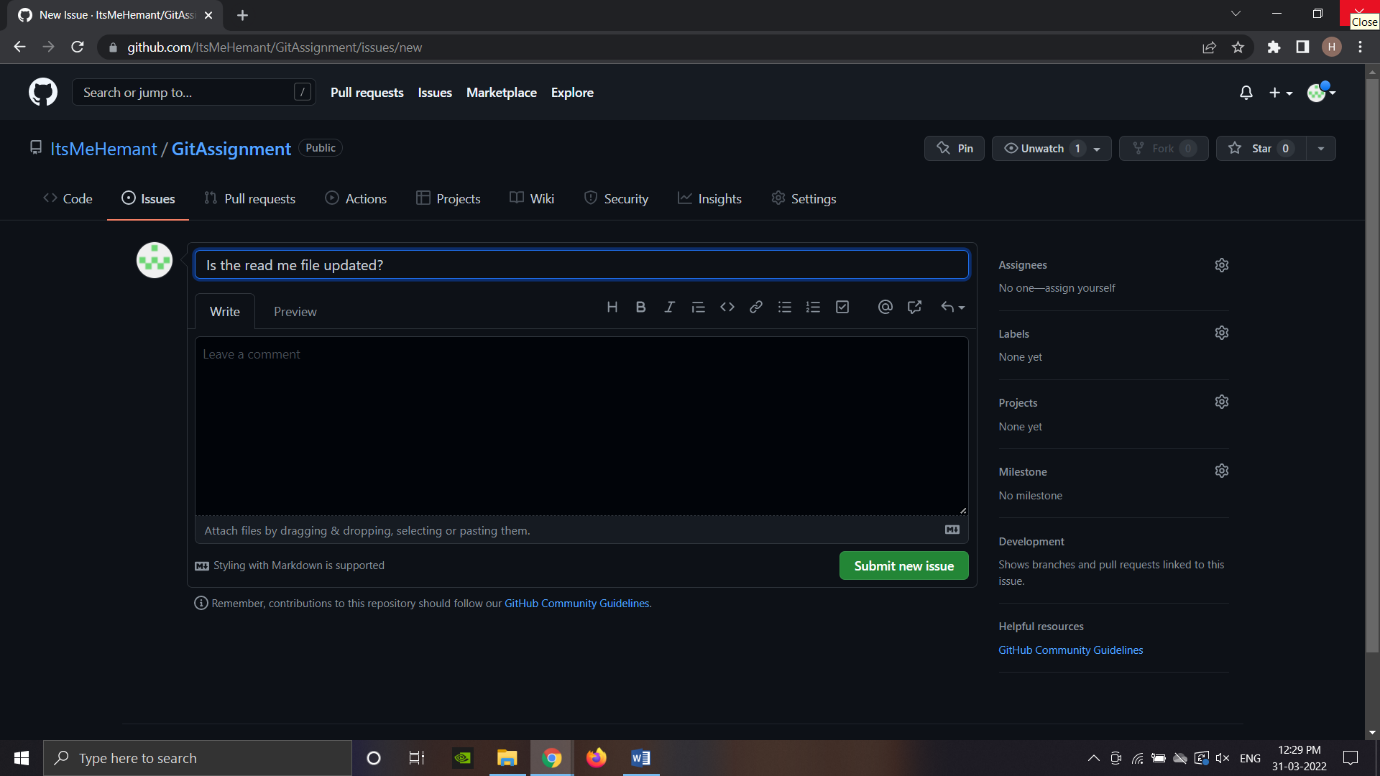
**7. Compare and merge the changes to the main branch.**



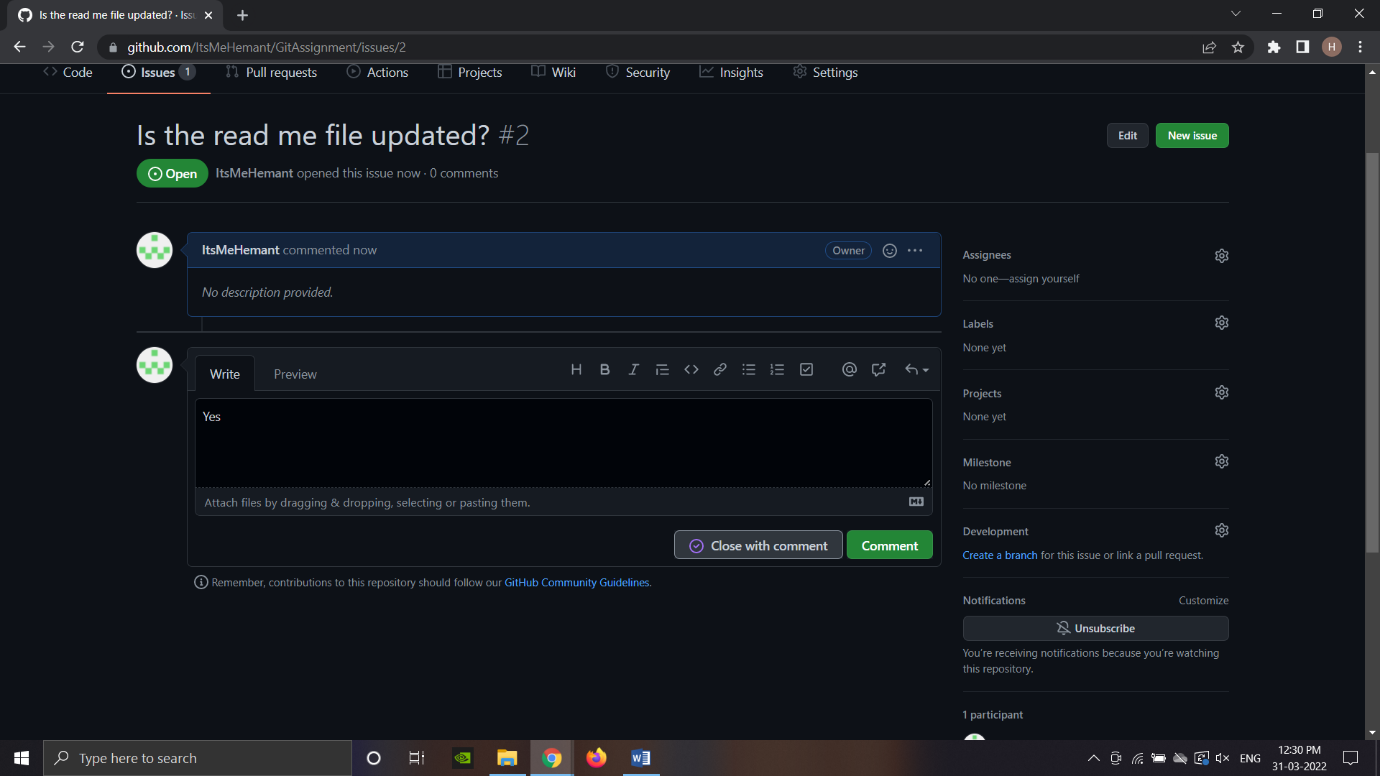
**8. Delete the edit-readme branch.**



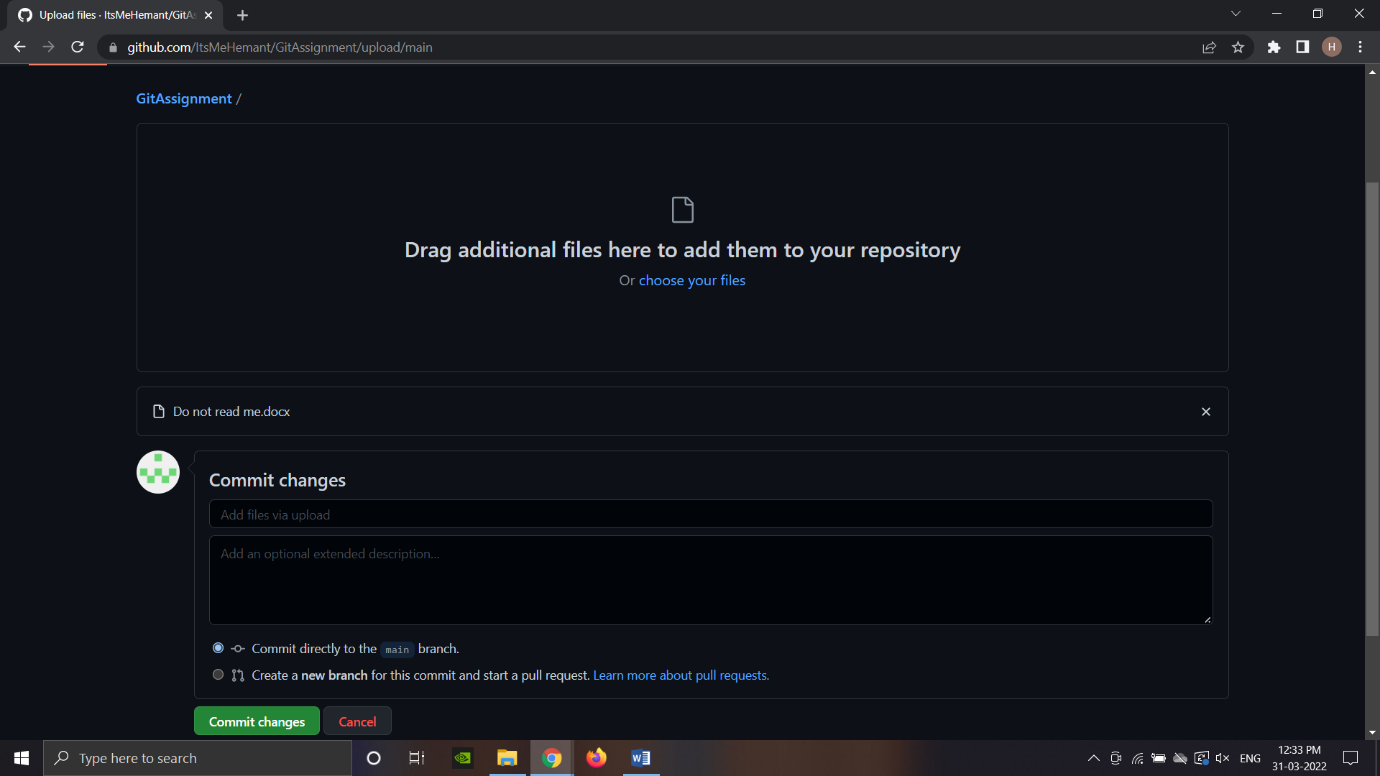
**9. Create a new issue and assign it to yourself/ any other contributor.**



**10. Preview the issue and close it.**



**11. Upload a file from your local drive to your Git repository.**



**Version Control Tool – Git**

**Assignment 2:**

**1. Configure Git Bash tool with the account credentials you created in the previous session.**

Hemant@LAPTOP-LO47BBP0 MINGW64 ~

$ git config --global user.name "ItsMeHemant"

Hemant@LAPTOP-LO47BBP0 MINGW64 ~

$ git config --global user.email "2004036.stu@dmscollege.ac.in"

**2. Create a new local Git repository in your project working directory.**

Hemant@LAPTOP-LO47BBP0 MINGW64 ~

$ mkdir myproject

Hemant@LAPTOP-LO47BBP0 MINGW64 ~

$ –cd myproject

Hemant@LAPTOP-LO47BBP0 MINGW64 ~/myproject

$ git init

Initialized empty Git repository in C:/Users/Hemant/myproject/.git/

**3. Create a .gitignore file and add the files not to be tracked.**

Hemant@LAPTOP-LO47BBP0 MINGW64 ~/myproject (master)

$ touch .gitignore

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

Hemant@LAPTOP-LO47BBP0 MINGW64 ~/myproject (master)

$ git commit -m "First release of Hello World"

[master (root-commit) 659bdde] First release of Hello World

3 files changed, 22 insertions(+)

create mode 100644 README.md

create mode 100644 index.html

create mode 100644 styles.css

**5. Modify a file and perform a second commit.**

Hemant@LAPTOP-LO47BBP0 MINGW64 ~/myproject (master)

$ git commit -a -m "Added a new line in index.html"

[master 8b8441c] Added a new line in index.html

1 file changed, 1 insertion(+)

**6. Display the repository status.**

Hemant@LAPTOP-LO47BBP0 MINGW64 ~/myproject (master)

$ git status

On branch master

nothing to commit, working tree clean

**7. Create a new branch.**

Hemant@LAPTOP-LO47BBP0 MINGW64 ~/myproject (master)

$ git branch hello-world-images

**8. Edit some files in the branch and merge the code with the master branch.**

Hemant@LAPTOP-LO47BBP0 MINGW64 ~/myproject (master)

$ git merge hello-world-images

Updating 8b8441c..a70a4c8

Fast-forward

image.jpg | Bin 0 -> 32861 bytes

index.html | 2 ++

2 files changed, 2 insertions(+)

create mode 100644 image.jpg

**9. Add the repository to github.com.**

Hemant@LAPTOP-LO47BBP0 MINGW64 ~/myproject (master)

$ git push --set-upstream origin master

Enumerating objects: 12, done.

Counting objects: 100% (12/12), done.

Delta compression using up to 8 threads

Compressing objects: 100% (12/12), done.

Writing objects: 100% (12/12), 33.09 KiB | 16.55 MiB/s, done.

Total 12 (delta 2), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (2/2), done.

To https://github.com/ItsMeHemant/hello-world.git

\* [new branch] master -> master

branch 'master' set up to track 'origin/master'.

**10. Learn to clone a repository.**

Hemant@LAPTOP-LO47BBP0 MINGW64 ~/myproject (master)

$ git clone https://github.com/ItsMeHemant/GitAssignment.git

Cloning into 'GitAssignment'...

remote: Enumerating objects: 15, done.

remote: Counting objects: 100% (15/15), done.

remote: Compressing objects: 100% (9/9), done.

remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 0

Receiving objects: 100% (15/15), 12.06 KiB | 12.06 MiB/s, done.