Practical no. 1a

(GIT)

1. Create an account in GitHub.

2. Create a repository, give a name to it and then thick the “initialize this repository with a Readme”.

3. Create repository.

4. Create new file and edit it and then commit changes.

5. Click on clone or download and copy link.

6. On your desktop create a folder and click on Git Clone.

7. Paste the link and then all your files will be cloned.

8. You can create new files and edit it. After editing, you have to commit changes and then push command and changes will be saved in your remote repository.

9. To get newly added files or changes in remote repository, you can click on tortoise git and then pull.

**Practical no. 1b**

**Git Merge**

1. Open GitHub.

2. Create a repository and create a file.

3. Type something and commit changes.

4. Add a collaborator by going in Setting/Collaborator and typing their username.

5. After your request accepted, your collaborator will fork your repository.

6. He will get an instance of your repository.

7. He can edit it and bring changes.

8. For merging both original file and forked file, you click on “Create Pull Request”.

9. You choose the files to be merged.

10. Confirm Merge.

11. If any conflicts happen, then by using resolve we can solve it.

12. After a conflict we click on resolve and both the files open and you can see the conflict and clear the problem.

**Practical no. 2**

**(User Stories)**

**Student**

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| Students should be able to borrow | **As a student**  **I want to** borrow the book  **So that** I can read the book. | Given: I am a registered Student  When: I see the name of book in the list of books  Then: I can borrow it. |

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| Students should be able to return | **As a student**  **I should be** **able** to return the book  **So that** I don’t get fined. | Given: I am a registered student  When: I want to return the book  Then: I should be able to return it and don’t get fined. |

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| Students should be able to search the book | **As a student**  **I want to** search the book  **So that I can** borrow the exact book. | Given: I am a registered student  When: I search for books name or accession number/author name and if the book is available  Then: I should be able to find the book. |

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| Students should be able to pay fine | **As a student**  **I should be** **able** to pay fine  **So I can** be able to borrow another book. | Given: I’m a registered student.  When: I have borrowed a book and I didn’t return the book on time  Then: I should be able to pay fine. |

**Librarian**

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| Librarian should be able to catalog the books. | **As a librarian**  **I should be able to** catalog the books  **So that** books can be searched easier. | Given: I am an official staff of college library  When: I want to catalog the book using the book’s type  Then: I should be able to catalog them accordingly. |

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| Librarian should be able to issue books. | **As a librarian**  **I should be able** to issue books  **So that** students needs shall be fulfilled. | Given: I am an official staff of college library  When: A student wants to borrow a book  Then: I should be able to issue the book. |

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| Librarian should be able to search for books. | **As a librarian**  **I want** to search for books  **So that** I can find the books and issue them to students faster. | Given: I am an official staff of college library  When: Students ask for a book  Then: I should be able to search it. |

**Online Shopping**

**Users: Customer, Admin**

**Admin**

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| Admin should be able to check stock | **As an admin**  **I should be able to** see the stock.  **So that** I can order new product if necessary. | Given: I am logged on as admin of the app  When: there is a shortage of required products  Then: I should be able check the stock and order new items. |

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| Admin should be able to send offer to customers. | **As an admin**  **I should be able to** give new offers to customers  **So that** I can get more attention for the website. | Given: I am admin of the app  When: there is need to attract more customers  Then: I send offers to customers. |

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| Admin should be able to add products | **As an admin**  **I want to** add products **so that** customers can see and buy products | Given: I am logged on as admin of the app  When: there is a new product in stock  Then: I will add the product to the website. |

**Customer**

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| A customer should be able to order items. | **As a customer**  **I should be** able to order items  **So that I** can get them delivered to me. | Given: I am logged in as customer  When: I see items online  Then: I should be able to order them. |

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| A customer should be able to return items, if items are not in desired condition. | **As a customer**  **I should be** able to return the delivered items if they are not in working condition  **So that** I can order new items. | Given: I am a registered customer of the app  When: The delivered items are not in desired condition  Then: I should be able to return them. |

**Attendance System (Student)**

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| A student should be able to view the attendance report (percentage) at the end of each month. | **As a student**  **I want to** see my attendance  **So that I** don’tfall short of the required attendance. | Given: I am a registered student of college  When: I want to check my attendance  Then: I should be able to check my attendance and make sure it’s above 75%. |

**Examination System**

***Student***

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| A student should be able to print hall-ticket for exam from the examination hall. | **As a student**  **I should be** able to get a hall-ticket  **So that I** can attend exams | Given: I am registered student of the college  When: I want to attend an exam  Then: A hall-ticket should be issued to me. |

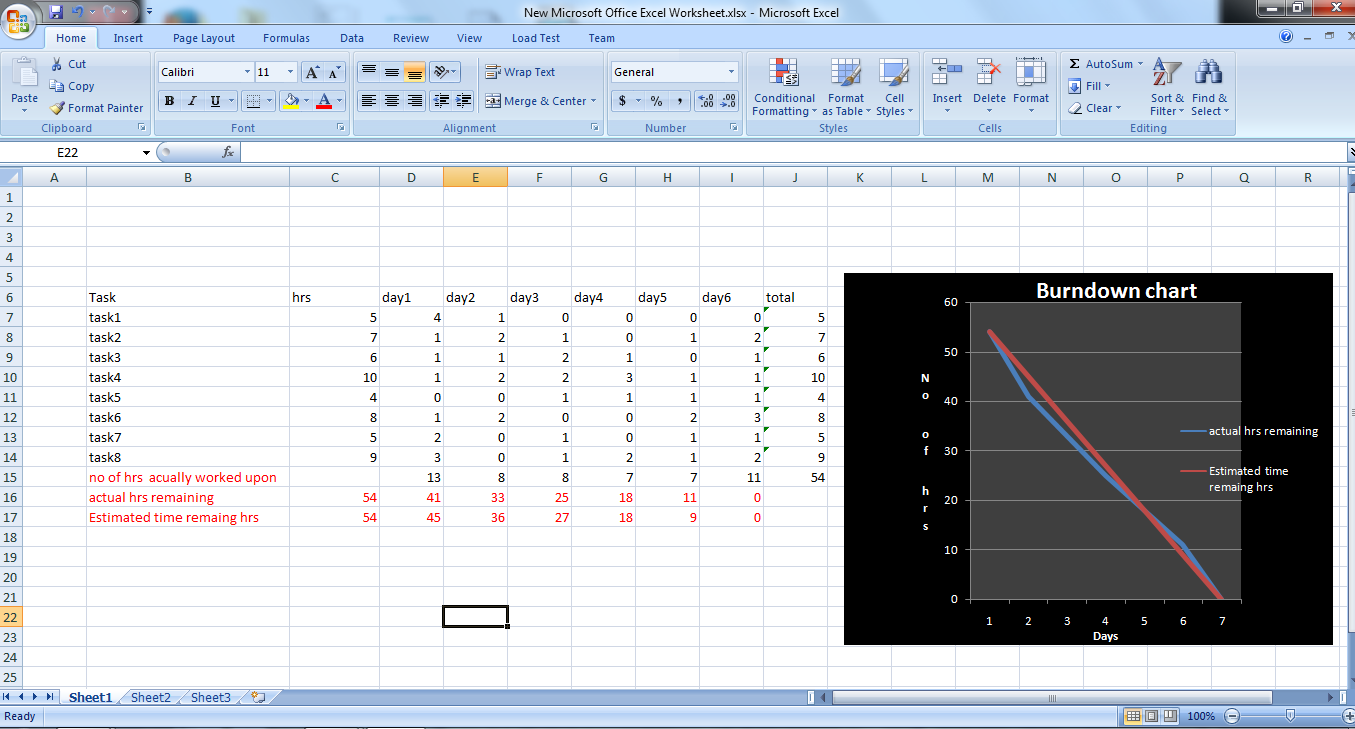
|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| A student should be able to view the supplementary exams marks. | **As a student**  **I want to** be allowed to see my supplementary exam results  **So that** I can check if I can proceed to next semester or no. | Given: I am registered student of the college  When: I appear for supplementary exam  Then: I should be able to check my result. |

***Faculty***

|  |  |  |
| --- | --- | --- |
| **Requirements:** | **User Story** | **Acceptance Criteria** |
| A faculty should be able to enter the marks of the students. | **As a faculty**  **I want to** enter the examination marks of students.  **So that** student’s report can be generated. | Given: I am an official faculty of the college  When: I have corrected the students papers  Then: I should be able to enter their marks. |

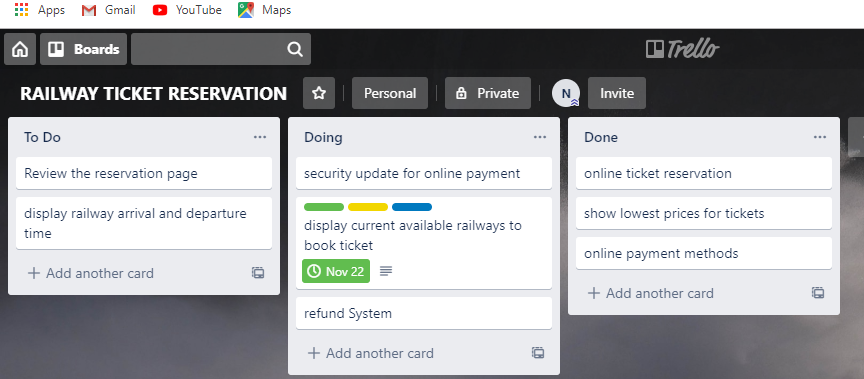
**Practical no. 3**

***(BURNDOWN CHARTS)***



**Practical no. 4**

**Trello**



**Practical no. 5**

**(JUNIT TEST)**

1. Open eclipse and create new project.

2. Create a new package in the created project.

3. Create a class.

4. Write your function in your newly created class and return.

5. Create a JUnit test case (Right click on the package and select the junit test case).

6. Instantiate an object of your class.

7. Test your result.

**PROGRAM:**

**import** **static** org.junit.Assert.\*;

**import** org.junit.Test;

**public** **class** cubeTest {

@Test

**public** **void** test() {

cube r1=**new** cube();

r1.setedge(3);

**int** res=r1.area();

*assertEquals*(54, res);

**int** resu=r1.valume();

*assertEquals*(27, resu);

}

}

**public** **class** cube {

**int** a;

**int** getedge(){

**return** (a);

}

**void** setedge(**int** b)

{

a=b;

}

**int** area(){

**return** (6\*a\*a);

}

**int** valume(){

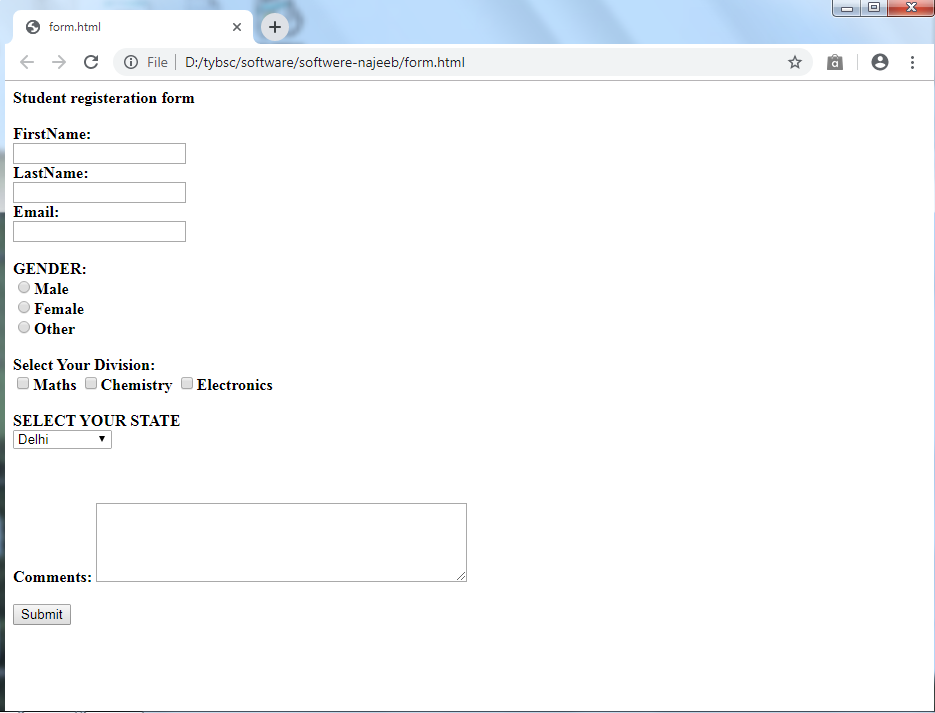
**return** (a\*a\*a);

}

}

**Practical no. 6**

**(SELENIUM)**

****

**package** src.selenium;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.support.ui.Select;

**public** **class** Html {

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

System.setProperty("webdriver.chrome.driver", "C:/selenium/webdriver/New folder/chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

driver.get("file:///C:/Users/pc/Desktop/form.html");

driver.findElement(By.name("firstname")).sendKeys("Masoud");

driver.findElement(By.name("lastname")).sendKeys("ghafary");

driver.findElement(By.name("email\_addreess")).sendKeys("masoud.ghafary@yahoo.com");

WebElement male = driver.findElement(By.name("gender1"));

male.click();

WebElement maths = driver.findElement(By.name("maths"));

maths.click();

WebElement chemistry = driver.findElement(By.name("chemistry"));

chemistry.click();

Select select = **new** Select(driver.findElement(By.name("select")));

select.selectByVisibleText("Goa");

driver.findElement(By.name("Description")).sendKeys("nen kfc ta zooooo aw dawood kharcha kawe aw topper di, d dawood ania woghayom");

driver.findElement(By.name("submit")).click();

driver.wait(200);

}

}

**Practical no. 7**

**(Refactoring Exercises)**

1. Rename: You can rename a variable, method or a class by using the rename command of refactoring.

Ex (for variable): Select the variable 🡪right click🡪refactor🡪rename 🡪 all variable with the same name will be updated.

2. Extract: You can convert a code into a method and a class.

Ex (code 🡪 method): Select code 🡪 right click 🡪 extract method (alt+shift+m) 🡪 give method name

Ex: to create a class automatically for variables: - right click on program 🡪 refactor 🡪 extract class

3. Inline: Convert a method to code.

Ex (method 🡪 code): Right click on method 🡪 inline (alt+shift+i)

**Practical no. 8**

**(Debugging)**

1. Write a program.

2. Right click on left side of code.

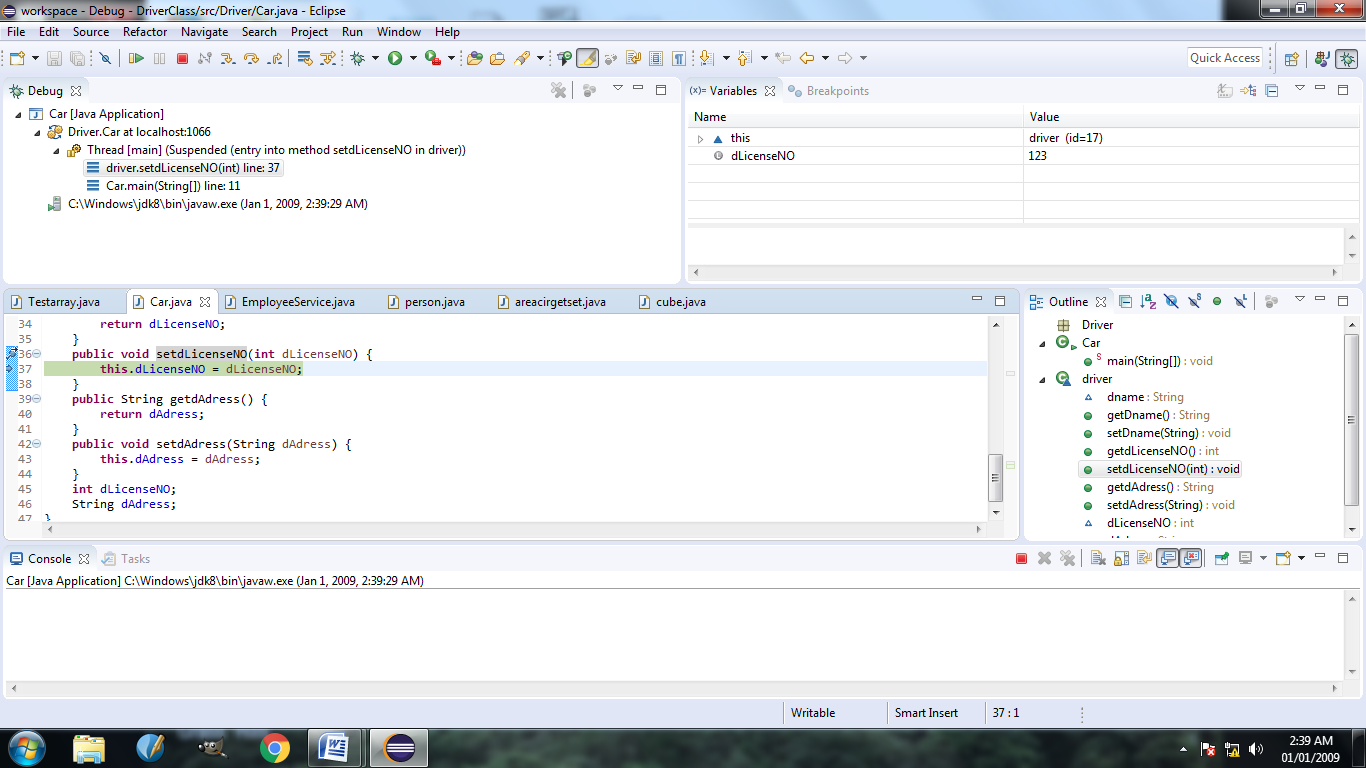
3. Select toggle break.

4. Right click on the program.

5. Debug as java application.

6. Press F5 for step-in.

7. Press F6 for step-out.



**Practical no. 9**

**(Maven Build)**

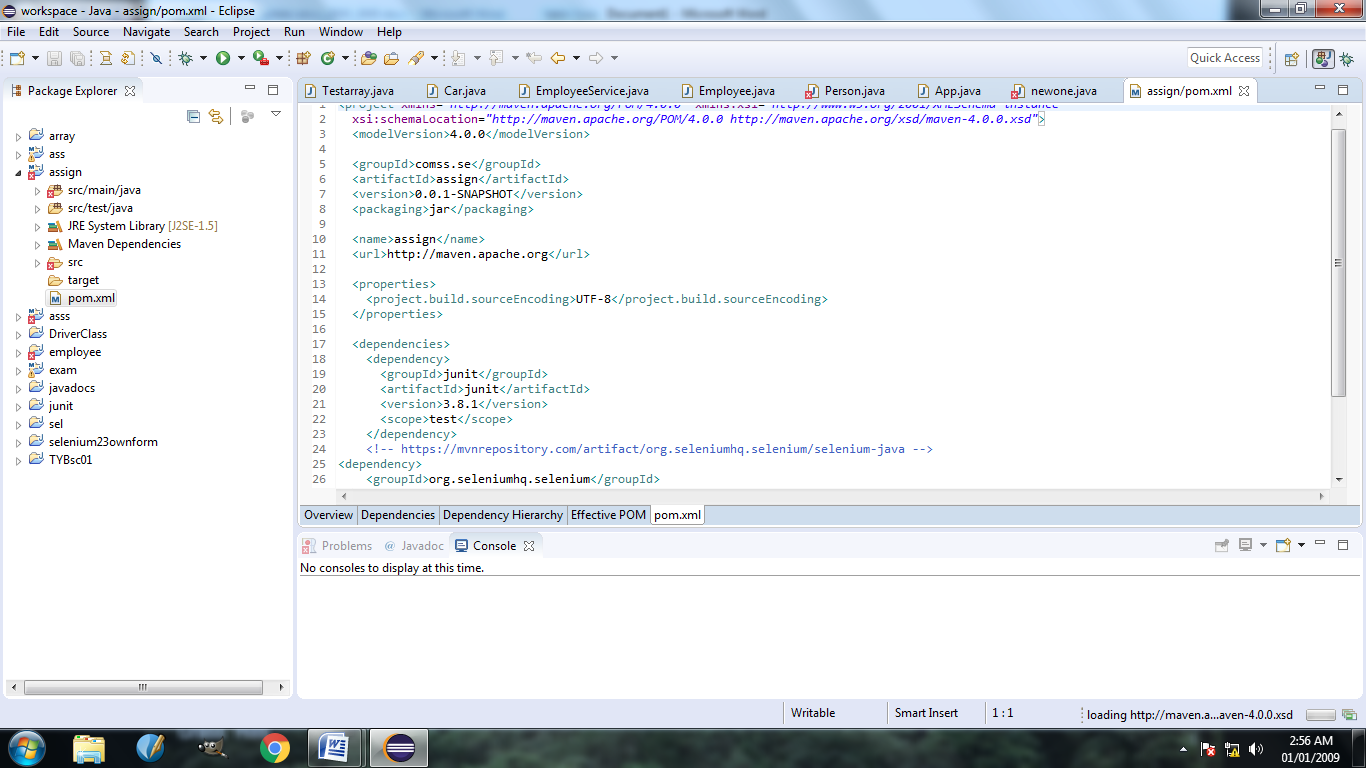
Maven: It downloads selenium jar files automatically from internet.

1. Open eclipse 🡪 create project 🡪 maven project 🡪 maven archetyp-quickstart

2. Give a name for Group id and Artifact id.

3. Right click on artifact id name 🡪 maven 🡪 update project

4. Open pom.xml, copy the latest dependencies from maven repositories and paste it between dependencies.



**Practical no. 10**

**(JAVADOC)**

1. Go to the program

2. Type /\*\*\* in a specific method or header where we want documentation.

