# Course Handout (Version 1.1) | Last update on 14th Dec, 2022

|  |  |  |  |
| --- | --- | --- | --- |
| Institute/ Name | Chitkara University Institute of Engineering & Technology | | |
| Department Name | Department of Computer Science & Engineering | | |
| Programme Name | Bachelor of Engineering (B.E.) Computer Science & Engineering | | |
| Course Name | Source Code Management | Session | 2022-23 |
| Course Code | CS181 | Semester/Batch | 2nd/2022 |
| L-T-P (Per Week) | 2-0-0 | Course Credits | 03 |
| Course Coordinator | Dr. Sunil Gupta | | |

1. **Scope and Objectives of the Course**

This course helps learners to become functional in open-source ecosystem. The course focus on enabling learners to examine the functionality of Software Version Control Systems. Version control systems are used to maintain various versions of same source code for maintainability and agility. The learners shall utilize the functionality of GIT to support version control of source code. The users of Software version control system are able to assess workflows in various version control systems like Git. The learners can apply the workflows to create collaboration with Co Participants on a software project.

After the completion of course student will be able to:

* Become familiar with environment of version control system.
* Utilize the functionalities of version control system to track running history for maintainability and agility of source code.
* Work in Collaborative teams to work through open-source ecosystem.

1. **Course Learning Outcomes**

On completion of the course, the student will be able to:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Course learning Outcome** | **POs** | **CL** | **KC** | **Sessions** |
| **CLO01** | Describe the fundamentals of source code management and its history with examples. | PO1,PO2,PO4,PO5,PO11,PO12 | K2 | Factual Conceptual | **5** |
| **CLO02** | Relate to best practices to be adopted by organizations to achieve continuous integration. | PO1,PO2,PO3,PO4,PO5,PO9,PO10,PO12 | K3 | Fundamental Conceptual | **5** |
| **CLO03** | Compare the utility of centralized and distributed version control systems and their basic operations. | PO1, PO2,PO5,PO9,PO12 | K4 | Conceptual Procedural | **4** |
| **CLO04** | Utilize distributed version control systems over centralized version control systems. | PO1,PO3,PO4,PO9,PO11,PO12 | K4 | Conceptual Procedural | **4** |
| **CLO05** | Design a remotely controlled repository in an open-source environment. | PO1,PO2,PO3, PO5,PO9,PO11, PO12 | K3 | Conceptual Procedural | **6** |
| **Total Sessions** | | | | | **30** |

Revised Bloom’s Taxonomy Terminology

\*Cognitive Level =CL

\*Knowledge Categories = KC

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Learning Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** |
| CLO01 | H | M |  | L | M |  |  |  |  |  | M | M |
| CLO02 | H | H | H | M | M |  |  |  | H | M |  | H |
| CLO03 | M | M |  |  | H |  |  |  | H |  |  | M |
| CLO04 | L |  | M | H |  |  |  |  | M |  | H |  |
| CLO05 | H | H | H |  | M |  |  |  | H |  | M | M |

H=High, M=Medium, L=Low

1. **ERISE Grid Mapping**

|  |  |
| --- | --- |
| **Feature Enablement** | **Level(1-5, 5 being highest)** |
| Entrepreneurship | 3 |
| Research | 4 |
| Innovation | 4 |
| Skills | 5 |
| Employability | 4 |

1. **Recommended Books (Reference Books/Text Books):**

**B01:** Pro Git by Scott Chacon and Ben Strob,Apress Available at https://git-scm.com/book/en/v2

**B02:** Learn Version Control with Git: A step-by-step course for the complete beginner by Tobias Günther

**B03:** GIT: The Ultimate Guide for Beginners: Learn Git Version Control by Jameson Garner

1. **Other readings and relevant websites:**

|  |  |
| --- | --- |
| **Serial No** | **Link of Journals, Magazines, websites and Research Papers** |
|  | <https://docs.github.com/en> |
|  | <https://docs.github.com/en/discussions> |
|  | <https://docs.github.com/en/communities> |
|  | <https://docs.github.com/en/developers> |
|  | Github.com |

1. **Recommended Tools and Platforms**

Github.com, Git Client available at <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

1. **Course Plan:**

|  |  |  |
| --- | --- | --- |
| **Session Number** | **Topics** | Recommended Book |
| 1-3 | Introducing Version Control – Git client(CLI, GUI), Linux environment Emulation  Installing git CLI and git GUI Initializing the repository, and exploring git –help | B01 |
| 4-6 | Exploring Github and Creating a Public Repository – Creating repository, understanding controls on the panel, working on Git Hub alone, realizing the significance of Git Client for Github utilization | B01 |
| 7-9 | Working With Git – Commands for initiating repos, managing repos  Git status, add, commit, stage – Life cycle of a file in Git managed in Repos  Git branches and HEAD, Git branches management, Create a new branch, Commit changes in the new branch, Explore commit in the new branch | B01,B02 |
|  | **Task 1.1 (Assessments)** |  |
| 10-12 | Git Cloning, exploring and modifying public repositories - Cloning repository, Exploring contents of the cloned repository, Unpacking Git objects, Exploring cloned repository in GitHub Desktop, Commit changes in the cloned repository  Git Configuration Files – creating personalized configurations | B02 |
|  | **Project work allocation : Students Project Group Initialize** |  |
| 13-15 | Git attributes and gitignore, Staging files - /attributes for managing, filtering, masking  Working With Git History – Forensics on GIT logs  Log, graphical history, undo changes in history – creating presentable GUI for GIT activity in versioned repos | B03 |
| 16-18 | Merge Resolution In Git – Branching, tagging branches, creating test, dev, prod branches Scenario creation for conflict creation while merging branches by a single user, multiple users | B01,B03 |
| 19-21 | Git branch, basic conflict and merge resolution workflow- Resolution of merge conflicts  GitHub and remote repositories - Cloning remote repository, What is a remote repository | B02,B03 |
| 22-24 | Git push, fetch and pull operations - Pushing to the remote repository, FETCH\_HEAD, perform a git pull, Git pull with fast forward merge, Resolving conflicts during Git pull | B02,B03 |
|  | **Task 1.2 - (Assessments)** |  |
| 25-30 | Project with teamwork demonstrating all aspects of GIT | B01,B02 |
|  | **Task 2 - Assessments : Project Completion of each group and Submission the project report.** |  |

1. **Delivery/Instructional Resources**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Session Number** | **Topics** | **PPT**  *(link of ppts on the central server)* | **Industry Expert Session** | **Web References** | **Audio-Video** |
| 1-3 | Introducing Version Control – Git client(CLI, GUI), Linux environment Emulation  Installing git CLI and git GUI Initializing the repository, and exploring git –help | [https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository" \t "_blank) |  | 1. <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git> | <https://git-scm.com/video/what-is-version-control> |
| 4-6 | Exploring Github and Creating a Public Repository – Creating repository, understanding controls on the panel, working on Git Hub alone, realizing the significance of Git Client for Github utilization | [https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository" \t "_blank) |  | 1. <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup> 2. <https://git-scm.com/book/en/v2/Getting-Started-Getting-Help> | <https://git-scm.com/video/what-is-git> |
| 7-9 | Working With Git – Commands for initiating repos, managing repos  Git status, add, commit, stage – Life cycle of a file in Git managed in Repos  Git branches and HEAD, Git branches management, Create a new branch, Commit changes in the new branch, Explore commit in the new branch | [https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository" \t "_blank) |  | 1. <https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository> 2. <https://git-scm.com/book/en/v2/GitHub-Account-Setup-and-Configuration> 3. https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes | <https://git-scm.com/video/what-is-version-control> |
| 10-12 | Git Cloning, exploring and modifying public repositories - Cloning repository, Exploring contents of the cloned repository, Unpacking Git objects, Exploring cloned repository in GitHub Desktop, Commit changes in the cloned repository  Git Configuration Files – creating personalized configurations | [https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository" \t "_blank)  [repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository" \t "_blank) |  | 1. <https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository> 2. <https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository> | https://git-scm.com/video/what-is-version-control |
| 13-15 | Git attributes and gitignore, Staging files - /attributes for managing, filtering, masking  Working With Git History – Forensics on GIT logs  Log, graphical history, undo changes in history – creating presentable GUI for GIT activity in versioned repos | [https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository" \t "_blank) |  | 1. <https://git-scm.com/book/en/v2/Git-Tools-Interactive-Staging> 2. <https://git-scm.com/book/en/v2/Git-Tools-Stashing-and-Cleaning> | https://git-scm.com/video/what-is-version-control |
| 16-18 | Merge Resolution In Git – Branching, tagging branches, creating test, dev, prod branches Scenario creation for conflict creation while merging branches by a single user, multiple users | [https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository" \t "_blank) |  | 1. <https://git-scm.com/book/en/v2/Customizing-Git-Git-Configuration> 2. <https://git-scm.com/book/en/v2/Customizing-Git-Git-Attributes> | https://git-scm.com/video/what-is-version-control |
| 19-21 | Git branch, basic conflict and merge resolution workflow- Resolution of merge conflicts  GitHub and remote repositories - Cloning remote repository, What is a remote repository | [https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository" \t "_blank) |  | 1. <https://git-scm.com/book/en/v2/Customizing-Git-Git-Attributes> | https://git-scm.com/video/what-is-version-control |
| 22-24 | Git push, fetch and pull operations - Pushing to the remote repository, FETCH\_HEAD, perform a git pull, Git pull with fast forward merge, Resolving conflicts during Git pull | [https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository" \t "_blank) | Industry expert talk on Git | 1. https://git-scm.com/book/en/v2/Git-Basics-Viewing-the-Commit-History | <https://git-scm.com/video/what-is-version-control> |
| 25-30 | Project work with teamwork demonstrating all aspects of GIT like opening and closing a pull request, collaborative work on GitHub, Explain utility of Fetch and Pull while collaboration and implement a distributed workflow in a team. | [https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository" \t "_blank) |  | NA |  |

1. **Action plan for different types of learners**

|  |  |  |
| --- | --- | --- |
| **Slow Learners** | **Average Learners** | **Fast Learners** |
| * Remedial Classes on Saturdays * Encouragement for improvement using peer tutoring * Use of audio and visual material * Use of Real-Life Examples | * Workshops * Formative Exercises used to highlight concepts and notions * E-notes and E-exercise to read in adaptive to pedagogic metrical. | * Coding Competitions * Design Solutions for complex problems * Presentation on topics beyond those covered in CHO |

1. **Evaluation Scheme & Components:Continuous evaluation shall be adopted**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evaluation Component** | **Type of Component** | **No. of Assessments** | **Weightage of Component** | **Mode of Assessment** |
| Component 1 | Task 1 | 02\* | 60% | Online on GitHub |
| Component 2 | Group Task | 01\*\* | 40% | Online on GitHub |
| **Total** | | **100%** | | |

\* In 02assessments of Task 1, the ERP system will automatically calculate the average of assessment marks for evaluation to be incorporated in trail results.

\*\* There will be no end-term exams and at the end, only students will submit a report of their project in the form of a hard copy of the course.

1. **Details of Evaluation Components:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evaluation Component** | **Description** | **Syllabus Covered (%)** | **Timeline of Examination** | **Weightage (%)** |
| Task 1 | Task 1.1 | Upto 50% | Week 4 | 60% |
| Task 1.2 | 51% - 100% | Week 8 |
| Task 2 | Project Completion and Report Submission from the entire course | 100% | Week 10 | 40% |
| **Total** | | | | 100% |

\* As per Academic Guidelines minimum of 75% attendance is required to become eligible for continuous evaluation

**Evaluation Components**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Assessment** | **Timeline of Conduct** | **Total Marks** | **Description of Tasks for Evaluation**  **But not limited is** |
| Task 1.1 | Week 4 | 30 | 1. Setting up of Git Client, 2. Setting up GitHub Account, 3. Generate logs 4. Create and visualize branches 5. Git lifecycle description |
| Task 1.2 | Week 8 | 30 | 1. Add collaborators on GitHub Repo 2. Fork and Commit 3. Merge and Resolve conflicts created due to own activity and collaborators activity. 4. Reset and Revert |
| Task 2 | Week 13 | 40 | 1. Create a distributed Repository and add members in project team 2. Open and close a pull request. 3. Each project member shall create a pull request on a team members repo and close pull requests generated by team members on own Repo as a maintainer. 4. Publish and print network graphs 5. Submission of report in PDF carrying screenshots and detailed writing of steps taken to achieve all tasks. |

1. **Syllabus of the Course:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Session Number** | **Topic** | **No. of Lectures** | **Weightage %** |
| 1-3 | Introducing Version Control – Git client(CLI, GUI), Linux environment Emulation  Installing git CLI and git GUI Initializing the repository, and exploring git –help | 3 | 30 % |
| 4-6 | Exploring Github and Creating a Public Repository – Creating repository, understanding controls on the panel, working on Git Hub alone, realizing the significance of Git Client for Github utilization | 3 |
| 7-9 | Working With Git – Commands for initiating repos, managing repos  Git status, add, commit, stage – Life cycle of a file in Git managed in Repos  Git branches and HEAD, Git branches management, Create a new branch, Commit changes in the new branch, Explore commit in the new branch | 3 |
| 10-12 | Git Cloning, exploring and modifying public repositories - Cloning repository, Exploring contents of the cloned repository, Unpacking Git objects, Exploring cloned repository in GitHub Desktop, Commit changes in the cloned repository  Git Configuration Files – creating personalized configurations | 3 | 30% |
| 13-15 | Git attributes and gitignore, Staging files - /attributes for managing, filtering, masking  Working With Git History – Forensics on GIT logs  Log, graphical history, undo changes in history – creating presentable GUI for GIT activity in versioned repos | 3 |
| 16-18 | Merge Resolution In Git – Branching, tagging branches, creating test, dev, prod branches Scenario creation for conflict creation while merging branches by a single user, multiple users | 3 |
| 19-21 | Git branch, basic conflict and merge resolution workflow- Resolution of merge conflicts  GitHub and remote repositories - Cloning remote repository, What is a remote repository | 3 |
| 22-24 | Git push, fetch and pull operations - Pushing to the remote repository, FETCH\_HEAD, perform a git pull, Git pull with fast forward merge, Resolving conflicts during Git pull | 3 |
| 25-30 | Project with teamwork demonstrating all aspects of GIT |  |  |

**This documentis approved by:**

|  |  |  |
| --- | --- | --- |
| **Designation** | **Name** | **Signature** |
| **Course Coordinator** | Dr. Sunil Gupta |  |
| **Head – Academic Delivery** | Dr. Navjeet Kaur |  |
| **Dean CSE** | Dr. Monit Kapoor |  |
| **Date** |  | |