UNIVERSITY BUEA (UB)				
FACULTY OF ENGINEERING AND TECHNOLOGY (FET)				
Programme	**Department**	**Specialty**	**Level**	**Sem.**
Bachelor of Engineering (BEng)	Computer Engineering		300	1

Course Code	**Course Title**	**Credit Value**	**L-T-P**
CEF345	Software Development Tools	4	*00-00-60*

Course Instructor	Dr. Valery Nkemeni	**Email:** nkemeni.valery@ubuea.cm

COURSE SYLLABUS

COURSE OBJECTIVES

- Introduce students to a variety of modern software development tools (IDEs, frameworks, version control systems, etc.).
- Encourage independent learning and collaboration through peer learning and project-based approaches.
- Equip students with hands-on experience in utilizing these tools to build software projects.
- Enhance presentation and communication skills through weekly presentations.

COURSE STRUCTURE

- **Peer Learning:** Students will research tools, share knowledge, and collaborate in small groups.
- **Project-Based Learning:** The course will revolve around completing weekly tasks and contributing to final projects.
- **Weekly Presentations:** Every student belonging to a group will present their task findings and project progress every week.

COURSE OUTCOMES

Upon completion of this course, the student will be able to:

- Understand the role of software development tools in the software development life cycle.
- Learn to install, configure, and use various Integrated Development Environments (IDEs).
- Understand version control systems and learn to use Git and GitHub.
- Learn to use Agile project management tools and techniques.
- Understand front-end and back-end development tools and frameworks.
- Learn to use database management systems and query languages.
- Understand testing and debugging tools and techniques.
- Learn to use DevOps tools and practices.
- Understand containerization using Docker.
- Apply software development tools to real-world projects.

COURSE PROGRESSION SHEET

Weeks	Type (L-T-P)	CHAPTER (LESSON) AND CONTENT
1	L	**Course Presentation**

2	P	**Week 1: Introduction to Software Development Tools** Overview of software development tools (IDEs, frameworks, version control). **Task:** - Form project groups (8 students per group). - Select a project idea (simple application development). - Set up GitHub accounts for each member - Create GitHub a repository for the group. **Presentation:** Each group presents their project idea and repository setup.
3	P	**Week 2: Version Control with Git and GitHub** Introduction to Git and GitHub for version control and collaboration. **Task:** - Practice basic Git commands (commit, push, pull, branching). - Practice collaboration by managing branches and pull requests. - Collaborate on a simple task using GitHub. **Presentation:** Demonstrate group collaboration via GitHub, showing commits, branches and pull request workflows
4	P	**Week 3: Agile Development Methodologies with Scrum** Introduction to Agile development methodologies with a focus on the Scrum framework. **Key Topics:** Scrum roles (Scrum Master, Product Owner), Sprint planning, Sprint retrospectives, Scrum board. **Task:** - Create a Scrum board using a tool like Taiga or any tool of your choice - Plan the first sprint for the project (define user stories, tasks, and assign roles). - Set up the Scrum framework for continuous sprint-based development. **Presentation:** - Groups will present their Scrum boards, Sprint Backlogs, and Sprint goals. - Explain the user stories and tasks selected for their first sprint.
5	P	**Week 4: Integrated Development Environment (IDE) - Visual Studio Code** Introduction to Visual Studio Code (setup, extensions, debugging). **Task:** - Set up VS Code, install relevant extensions (e.g., Python, JavaScript), and explore debugging features. - Create a simple program (e.g., Python or JavaScript) - Debug simple applications using the IDE. - Use Visual Studio Code to implement tasks from the Sprint Backlog. - **Presentation:** Demonstrate IDE setup and debugging in VS Code.
6	P	**Week 5: Building a Simple Web Application (HTML/CSS/JavaScript)** Basic web development with HTML, CSS, and JavaScript. **Task:**

		- Build a basic web page using HTML, CSS, and JavaScript. Deploy the page using GitHub Pages. **Presentation:** Groups present and demo their web page, explaining how they used Git and VS Code.
7	P	**Week 6: Database Management with MySQL** Introduction to MySQL and database tools like MySQL Workbench. **Task:** - Create a simple database with MySQL, practice SQL queries, and connect the database to the previously built web application (Week 5). - Integrate a MySQL database into the project (setting up tables, querying data). - Plan the next sprint for backend and database integration. **Presentation:** Demonstrate database creation, SQL queries, and integration with the web app.
8	P	**Week 7: Backend Development with Node.js (Express)** Introduction to Node.js and Express for backend development. **Task:** - Develop a simple backend for the web application using Node.js. - Build a basic REST API using Node.js and Express. - Add functionality to connect the API to the MySQL database created in Week 6. - Work on user authentication and database queries. **Presentation:** - Present backend progress and how it connects to the frontend. - Groups demonstrate their REST API and explain how they integrated it with the database.
9	P	**Week 8: Frontend Development with React.js** Introduction to React.js for building interactive frontend applications. **Task:** - Build the React.js frontend that integrates with the Node.js backend. - Create a simple React.js application that fetches data from the Node.js REST API (Week 7). **Presentation:** Demonstrate the React.js application with working frontend-backend communication
10	P	**Week 9: Testing and Debugging (Unit Testing & Automation)** Introduction to testing frameworks (e.g., JUnit for Java, PyTest for Python). **Task:** - Write unit tests for the previously developed projects (either backend or frontend) and set up automated testing using GitHub Actions. **Presentation:** Groups demonstrate their testing strategies and automated test pipelines.
11	P	**Week 10: Cloud Deployment with Docker** Introduction to Docker for containerizing applications and deploying them to the cloud.

		Task: - Containerize the web application (React frontend + Node.js backend) and deploy it using Docker. **Presentation:** Groups present their containerized application and explain the Docker setup.
12	P	**Week 11: Final Project Presentations and Evaluation** **Task:** Each group prepares a final presentation to showcase their complete project. **Presentation:** Groups present their final project, covering the development process, tools used, challenges faced, and lessons learned. **Final Project Submission:** Submit the completed project for evaluation.
L: Lecture	**T: Tutorials**	**P: Practical**

MODE OF ACCESSMENT				
Weekly presentations	Attendance	Individual Performance	Group Performance	**Total**
	2	3	2	7
Total Semester	**20**	**30**	**20**	**70**
Final Project Presentation				**30**
TOTAL				**100**