# UNIVERSITY OF MUMBAI



# Syllabus
# For the
# Program: F.Y.B.Sc. Sem -I CBCS
# Course: Computer Science

**(Choice Based and Credit System with effect from the academic year 2021-22)**

# Preamble

The rise of Information and Communication Technology (ICT) has profoundly affected modern society. Increasing applications of computers in almost all areas of human endeavor has led to vibrant industries with concurrent rapid change in technology.

As the computing field advances at a rapid pace, the students must possess a solid foundation that allows and encourages them to maintain relevant skills as the field evolves. Specific languages and technology platforms change over time. Thus students must continue to learn and adapt their skills throughout their careers. To develop this ability, students will be exposed to multiple programming languages, tools, paradigms and technologies as well as the fundamental underlying principles throughout this programme.

The programme offers required courses such as programming languages, data structures, computer architecture and organization, algorithms, database systems, operating systems, and software engineering; as well as specialized courses in artificial intelligence, computer-based communication networks, distributed computing, information security, graphics, human-computer interaction, multimedia, scientific computing, web technology, and other current topics in computer science.

The core philosophy of this programme is to –

- ☐ Form strong foundations of Computer Science
- ☐ Nurture programming, analytical & design skills for the real world problems.
- ☐ Introduce emerging trends to the students in gradual way.
- ☐ Groom the students for the challenges of ICT industry

The students these days not only aspire for a career in the industry but also look for research opportunities. The main aim of this programme is to deliver a modern curriculum that will equip graduates with strong theoretical and practical backgrounds to enable them to excel in the workplace and to be lifelong learners. Not only does it prepare the students for a career in Software industry, it also motivates them towards further studies and research opportunities. Graduating students, can thus take up postgraduate programmes in CS leading to research as well as R&D, can be employable at IT industries, or can adopt a business management career.

In the first year i.e. for semester I & II, basic foundation of important skills required for software development is laid. The syllabus proposes to have four core subjects of Computer science and two core courses of Mathematics-Statistics. All core subjects are proposed to have theory as well as practical tracks. While the Computer Science courses will form fundamental skills for solving computational problems, the Mathematics & Statistics course will inculcate research-oriented acumen. Ability Enhancement Courses on Soft Skill Development will ensure an overall and holistic development of the students. The syllabus design for further semesters encompasses more advanced and specialized courses of Computer Science.

We sincerely believe that any student taking this programme will get very strong foundation and exposure to basics, advanced and emerging trends of the subject. We hope that the students" community and teachers" fraternity will appreciate the treatment given to the courses in the syllabus.

We wholeheartedly thank all experts who shared their valuable feedbacks and suggestions in order to improvise the contents; we have sincerely attempted to incorporate each of them. We further thank Chairperson and members of Board of Studies for their confidence in us.

Special thanks to Department of Computer Science and colleagues from various colleges, who volunteered or have indirectly, helped designing certain specialized courses and the syllabus as a whole.

# Academic year 2021-2022

## Semester – I

| Course Code | Course Type | Course Title | Credits | Lectures/Week |
|---|---|---|---|---|
| USCS101 | Core Subject | Digital Systems & Architecture | 2 | 3 |
| USCSP101 | Core Subject Practical | Digital Systems & Architecture – Practical | 1 | 3 |
| USCS102 | Core Subject | Introduction to Programming with Python | 2 | 3 |
| USCSP102 | Core Subject Practical | Introduction to Programming with Python – Practical | 1 | 3 |
| USCS103 | Core Subject | LINUX Operating System | 2 | 3 |
| USCSP103 | Core Subject Practical | LINUX Operating System – Practical | 1 | 3 |
| USCS104 | Core Subject | Open Source Technologies | 2 | 3 |
| USCSP104 | Core Subject Practical | Open Source Technologies – Practical | 1 | 3 |
| USCS105 | Core Subject | Discrete Mathematics | 2 | 3 |
| USCSP105 | Core Subject Practical | Discrete Mathematics – Practical | 1 | 3 |
| USCS106 | Core Subject | Descriptive Statistics | 2 | 3 |
| USCSP106 | Core Subject Practical | Descriptive Statistics – Practical | 1 | 3 |
| USCS107 | Ability Enhancement Course | Soft Skills | 2 | 3 |

# Semester I

| Course Code | Course Title | Credits | Lectures /Week |
|---|---|---|---|
| USCS104 | **Open Source Technologies** | **2** | **3** |
| | | | |

**About the Course:**
Open Source Software is becoming an important resource for development, especially in developing countries. A working understanding of the economic and technical background of the Free / Open Source Software movement (FOSS) is essential for its effective use. The course takes students through the history and current status of the FOSS world, and starts them exploring it, by connecting their personal experiences with corresponding FOSS projects. Students will experience finding and using Open Source Software projects.

**Course Objectives:**
- Understand the difference between open-source software and commercial software.
- Understand the policies, licensing procedures and ethics of FOSS.
- Understand open-source philosophy, methodology and ecosystem.
- Awareness with Open-Source Technologies.

**Learning Outcomes:**
- Differentiate between Open Source and Proprietary software and Licensing.
- Recognize the applications, benefits and features of Open-Source Technologies
- Gain knowledge to start, manage open-source projects.

| Unit | Topics | No of Lectures |
|---|---|---|
| I | **Introduction to Open-Source:** Open Source, Need and Principles of OSS, Open-Source Standards, Requirements for Software, OSS success, Free Software, Examples, Licensing, Free Vs. Proprietary Software, Free Software Vs. Open-Source Software, Public Domain. History of free software, Proprietary Vs Open-Source Licensing Model, use of Open- Source Software, FOSS does not mean no cost. History: BSD, The Free Software Foundation and the GNU Project.<br><br>**Open-Source Principles and Methodology:** Open-Source History, Open-Source Initiatives, Open Standards Principles, Methodologies, Philosophy, Software freedom, Open-Source Software Development, Licenses, Copyright vs. Copy left, Patents, Zero marginal cost, Income-generation Opportunities, Internationalization.<br><br>**Licensing:** What Is A License, How to create your own Licenses, Important FOSS Licenses (Apache, BSD, PL, LGPL), copyrights and copy lefts, Patent. | 15 |

| | | |
|---|---|---|
| **II** | **Open-Source projects:** Starting and maintaining own Open-Source Project, Open-Source Hardware, Open-Source Design, Open-source Teaching, Open-source media.<br><br>**Collaboration:** Community and Communication, Contributing to Open-Source Projects Introduction to GitHub, interacting with the community on GitHub, Communication and etiquette, testing open-source code, reporting issues, contributing code. Introduction to Wikipedia, contributing to Wikipedia or contributing to any prominent open-source project of student's choice.<br><br>**Open-Source Ethics and Social Impact:** Open source vs. closed source, Open-source Government, Ethics of Open-source, Social and Financial impacts of open-source technology, Shared software, Shared source, Open Source as a Business Strategy | |
| **III** | **Understanding Open-Source Ecosystem**: Open-Source Operating Systems: GNU/Linux, Android, Free BSD, Open Solaris. Open-Source Hardware, Virtualization Technologies, Containerization Technologies: Docker, Development tools, IDEs, Debuggers, Programming languages, LAMP, Open-Source Database technologies<br><br>**Case Studies:** Example Projects: Apache Web server, BSD, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, WordPress, Git, GCC, GDB, GitHub, Open Office, LibreOffice<br>Study: Understanding the developmental models, licensing, mode of funding, commercial/non-commercial use. | **15** |

**Textbooks:**
1. "Open-Source Technology", Kailash Vadera&Bhavyesh Gandhi, University Science Press, Laxmi Publications, 2009
2. "Open-Source Technology and Policy", Fadi P. Deek and James A. M. McHugh, Cambridge University Press, 2008.

**Additional References**:
1. "Perspectives on Free and Open-Source Software", Clay Shirky and Michael Cusumano, MIT press.
2. "Understanding Open Source and Free Software Licensing", Andrew M. St. Laurent, O'Reilly Media.
3. "Open Source for the Enterprise", Dan Woods, GautamGuliani, O'Reilly Media
4. Linux kernel Home: http://kernel.org4
5. Open-Source Initiative: https://opensource.org/5
6. The Linux Foundation: http://www.linuxfoundation.org/
7. The Linux Documentation Project: http://www.tldp.org/2
8. Docker Project Home: http://www.docker.com3.
9. Linux Documentation Project: http://www.tldp.org/6
10. Wikipedia: https://en.wikipedia.org/7.https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia8
11. GitHub: https://help.github.com/9.
12. The Linux Foundation: http://www.linuxfoundation.org/

| Course Code | Course Title | Credits | Lectures /Week |
|---|---|---|---|
| **USCSP104** | **Open Source Technologies– Practical** | **1** | **3** |
| | | | |
| 1 | **Open Source Operating Systems**<br>☐ Learn the following open source operating system of your choice: Linux, Android, FreeBSD, Open Solaris etc.<br>☐ Learn the installation.<br>☐ Identify the unique features of these OS. | | |
| 2 | Hands with **LibreOffice**<br>☐ Learn it from practical view-point<br>☐ Give a brief presentation about it to the class | | |
| 3 | 'Hands on with **GIMP** Photo Editing Tool<br>☐ Learn it from practical view-point<br>☐ Give a brief presentation about it to the class | | |
| 4 | Hands on with **Shotcut** Video Editing Tool<br>☐ Learn it from practical view-point<br>☐ Give a brief presentation about it to the class | | |
| 5 | Hands on with **Blender** Graphics and Animation Tool<br>☐ Learn it from practical view-point<br>☐ Give a brief presentation about it to the class | | |
| 6 | Hands on with **Apache** Web Server<br>☐ Learn it from practical view-point<br>☐ Give a brief presentation about it to the class | | |
| 7 | Hands on with **WordPress** CMS<br>☐ Learn it from practical view-point<br>☐ Give a brief presentation about it to the class | | |
| 8 | Contributing to **Wikipedia**:<br>☐ Introduction to wikipedia: operating model, license, how to contribute?<br>☐ Create your user account on wikipedia<br>☐ c. Identify any topic of your choice and contribute the missinginformation | | |

| 9 | **Github**<br>    Create and publish your own open source project: Write any simple program using your choice of programming language.<br>    Create a repository on github and save versions of your project. You"ll learn about the staging area, committing your code, branching, and merging,<br>    Using GitHub to Collaborate: Get practice using GitHub or other remote repositories to share your changes with others and collaborate on multi- developer projects. You'll learn how to make and review a pull request on GitHub.<br>d. Contribute to a Live Project: Students will publish a repository containing their reflections from the course and submit a pull request. |
|---|---|
| 10 | **Virtualization:** Open Source virtualization technologies:<br>☐ Install and configure the following: VirtualBox, Zen, KVM<br>☐ Create and use virtual machines |
| 11 | **Containerization:**<br>☐ Install and configure the following containerization technologies: docker, rocket, LXD<br>☐ Create and use containers using it |

## Evaluation Scheme

I. **Internal Evaluation for Theory Courses – 25 Marks**

**(i) Mid-Term Class Test– 15Marks**

☐ It should be conducted using any **learning management system** such as **Moodle** (Modular object-oriented dynamic learning environment)

☐ The test should have **15 MCQ's** which should be solved in a time duration of **30 minutes.**

**(ii) Assignment/ Case study/ Presentations– 10 Marks**

☐ Assignment / Case Study Report / Presentation can be uploaded on any **learning management system**.

II. **External Examination for Theory Courses – 75 Marks**

☐ Duration: **2.5 Hours**

☐ Theory question paper pattern:

| | **All questions are compulsory.** | | |
|---|---|---|---|
| **Question** | **Based on** | **Options** | **Marks** |
| Q.1 | Unit I | *Any 4 out of 6* | 20 |
| Q.2 | Unit II | *Any 4 out of 6* | 20 |
| Q.3 | Unit III | *Any 4 out of 6* | 20 |
| Q.4 | Unit I,II and III | *Any 5 out of 6* | 15 |

☐ All questions shall be compulsory with internal choice within the questions.

☐ Each Question maybe sub-divided into subquestions as a, b, c, d, etc. & the allocation of Marks depends on the weightage of the topic.

III. **Practical Examination**

☐ Each core subjectcarries50 Marks

**40 marks + 05 marks (journal) + 05 marks (viva)**

☐ Duration: **2 Hours** for each practical course.

☐ Minimum **80% practical** from each core subjects are required to be completed. ☐ **Certified Journal is compulsory for appearing at the time of Practical Exam** ☐ The final submission and evaluation of **journal in electronic form** using a Learning Management System / Platform can be promoted by college.

--------------------------------