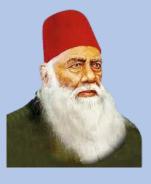


2023-24



جب کوئی قوم فن اور
علم سے عاری ہو جاتی
ہے تو وہ غربت کو
دعوت دیتی ہے اور
جب غربت آتی ہے تو
وہ ہزاروں جرائم کو
جنم دیتی ہے۔

"When a nation becomes devoid of art and learning, it invites poverty and when poverty comes it brings in its wake thousands of crimes."
-Sir Syed Ahmad
Khan

**Laboratory Course-III Course Code-CAMS3P01** 



# DEPARTMENT OF COMPUTER SCIENCE

ALIGARH MUSLIM UNIVERSITY, ALIGARH 2023-2024



The following lab manual up-gradation committee updated the lab manual:

	□ Prof. Aasim Zafar (Chairperson)
	🗖 Dr. Suhel Mustajab
	🗖 Dr. Arman Rasool Faridi
	🗖 Dr. Faisal Anwar
	□ Dr. Mohammad Nadeem
The following c	ommittee member originally design the lab manual:
	🗖 Prof. Mohammad Ubaidullah Bokhari
	🗖 Dr. Arman Rasool Faridi
	Dr. Faisal Anwer
	□ Prof. Aasim Zafar (Converner)
Design L Comp	vilation:
	🗖 Dr. Shakeel Ahamad

Revised Edition: June, 2023

Department of Computer Science, A.M.U., Aligarh (U.P.) India

# **Lab Manual: Laboratory Course-III (CAMS-3P01)**

CONTENTS	PAGE NO
COURSE DESCRIPTION	2
CONTENT	2
OBJECTIVES	3
OUTCOMES	4
HOW TO DO WELL IN THIS COURSE	5
TEACHING METHODS AND ASSESMENTS	7
SUBMISSION OF DELIVERABLES	9
EVALUATION AND GRADING	9
REQUIRED TEXTS/READING/REFERENCES	10
ATTENDANCE POLICY	10
LATE POLICY	10
MILESTONES FOR MINI-PROJECT	11
TOPICAL CALENDAR	14
WEEK-WISE LAB ASSIGNMENT BASED ON SUBJECTS	15

COURSETITLE: Laboratory Course-III COURSE CODE: CAMS – 3P01

CREDIT: 04 PERIODS PER WEEK: 06

CONTINUOUS ASSESSMENT: 40 Marks EXAMS: 60 Marks

#### **COURSE DESCRIPTION**

Application of classroom knowledge and skills in computer science to solve real-world problems and to develop research and software development skills.

#### **CONTENT**

This course consists of the development of a realistic application, representative of a typical real-life software system, under semi-professional working conditions. The students are expected to propose, analyses, design, develop, test and implement asoftware system. The student will deliver oral presentations, progress reports, and a final report.

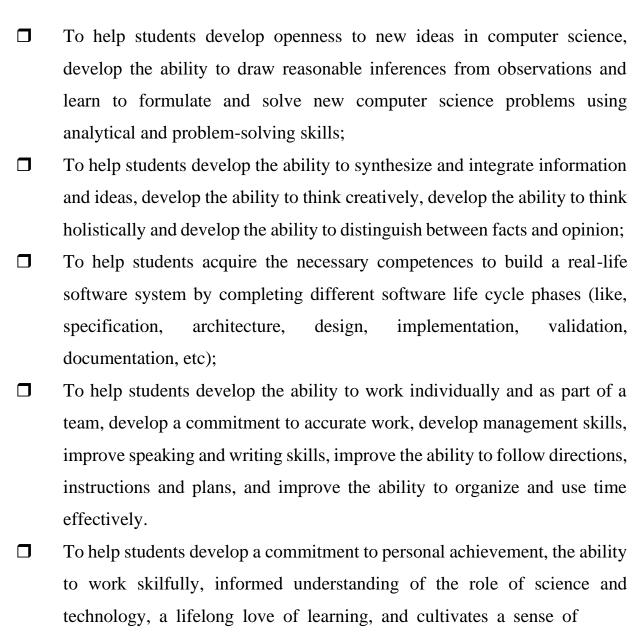
Depending on the topic of the project and the chosen software development methodology, which may vary from one year to another, the following themes may be addressed to some extent:

- Software development methodologies, static (products) and dynamic aspects (processes);
- Requirement analysis (goals, use cases), software architectures, architectural styles and patterns, model-driven engineering (MDE);
- Programming techniques, software development environments, refactoring;
- Software validation through unit tests, integration tests, functional and structural tests, and code reviews.
- Project management, planning, resource estimation, reporting.
- Version management by using a version management tool.
- Examples of kinds of systems to be developed are distributed systems,
   client/server systems, web-based systems, secure systems, mobile systems,

adaptable systems, optimizations of existing systems or data-intensive systems, etc.

Besides completing a mini project, the students are required to complete subject related Lab Assignments given by respective course teachers. The individual teachers who are teaching the courses with lab component will be responsible for giving assignments, monitoring and evaluating their respective assignments.

#### **OBJECTIVES**



responsibility for one's own behaviour and improves self-esteem/self-confidence.

# **OUTCOMES**

Upon successful completion of this course students will be able to:

Identify project/research problems; understand information and grasp
meaning; translate knowledge into new context; use information, methods,
concepts, and theories of fundamental topics in computer science in new
situations (Knowledge, Comprehension);
Apply computer science principles and practices to a real-world problem;
demonstrate in-depth knowledge in the area of the project they have
undertaken; solve problems using required knowledge and skills; implement
and test solutions/algorithms (Application and Evaluation);
Identify potential solutions/algorithms for the project problem; see patterns
and modularize the problem, recognize hidden meanings and identify
components, show proficiency in software engineering principles
(Analysis);
Apply a software development methodology currently practiced in industry
to produce software system in a rigorous and systematic way using different
software life cycle phases (specification, architecture, design,
implementation, validation, documentation) (Synthesis);
Show evidence (group collaboration, regular meetings, email
communications, significant knowledge and skills contributions, etc.) of
working productively as an individual and in a team on a project that
produces a significant software product ( <i>Team Work</i> );
Show evidence of competency in oral and written communications skills
through oral presentations (project presentation, department seminar or

conferences, client interactions), technical reports and/or published research
papers in conferences and/or journals (Communications);
Use modern techniques, skills and tools necessary for computer science
practices relevant to the project they undertake: use techniques in recent

research papers to solve problems (Lifelong Learning).

# HOW TO DO WELL IN THIS COURSE

The students are advised to attend all their theory classes and respective labs
regularly as both are integrated to each other. If any student will miss the theory
lecture, he/she may not able to do well in lab related to that topic.
The students are advised to submit the assignments given in theory and lab classes timely to their respective Teachers/Instructors.
The students should demonstrate disciplined and well behaved demeanor in the Department.
Each student shall be assigned a system in their introductory lab. They are advised to do their work on that system only for the whole semester. Students should store all their lab activities regularly.
All students are advised to understand course objectives and outcomes and achieve both during their lab work.
The students are advised to follow books/eBooks/online tutorial/other online study material links given in lecture/lab manual/ syllabus references. These study materials are very helpful in terms of skills, knowledge and placement.
This lab course is very important in terms of placement. Therefore, students are advised to implement all the problems by her/him given in the individual week.

☐ All students are advised to solve old placement papers for campus selection. Following links may be useful for the preparation of your campus placements. https://www.indiabix.com/placement-papers/companies/ https://www.offcampusjobs4u.com/download-tcs-placement-testquestion-papers-with-solutions/ https://www.indiabix.com/placement-papers/tcs/ https://www.firstnaukri.com/career-guidance/infosys-placement-paperswith-solutions-2019-firstnaukri-prep https://prepinsta.com/ibm/ ➤ https://www.faceprep.in/infosys/infosys-aptitude-questions/ https://alpingi.com/infosys-placement-papers-solution-pdf-download/ ➤ http://placement.freshersworld.com/ The Students are advised to follow below tutorials' links: https://www.w3schools.com/php/php\_intro.asp https://www.w3schools.com/js/default.asp https://www.w3schools.com/sql/default.asp https://www.tutorialspoint.com/php/index.htm > https://www.geeksforgeeks.org/php/ https://www.geeksforgeeks.org/sql-tutorial/ https://www.geeksforgeeks.org/javascript-tutorial/ The Students are advised to follow below Links for installing application software: https://www.eclipse.org/pdt/ http://php.net/manual/en/install.php https://docs.microsoft.com/en-us/dotnet/framework/install/guide-fordevelopers https://httpd.apache.org/download.cgi

- https://docs.microsoft.com/en-us/sql/database-engine/install-windows/install-sql-server?view=sql-server-ver15
- https://www.sqlservertutorial.net/install-sql-server/
- ☐ The Students are advised to follow below Links for online compilers :
  - ► <a href="https://www.onlinegdb.com/online\_php\_interpreter">https://www.onlinegdb.com/online\_php\_interpreter</a>
  - http://phpfiddle.org/
  - https://www.tutorialspoint.com/execute\_sql\_online.php
  - https://www.onlinegdb.com/online\_sqlite\_editor
  - https://www.onlinegdb.com/online\_csharp\_compiler
- ☐ Skill Set that are required to develop by the students of MCA:
  - ➤ Good communication and behavioral skills
  - ➤ A positive attitude
  - **≻** Confidence
  - > Strong technical skills
  - ➤ Good command over programming languages like C, C++, Java, .Net, etc.
  - ➤ Good programming skills and hands on experience
  - ➤ Knowledge of data structure and database
  - ➤ Awareness of latest technology trends.

# TEACHING METHODS AND ASSESMENTS FOR ACHIEVING LEARNING OUTCOMES

This lab class will meet thrice per week for 100 minutes each meeting -- some lab class time may be traditional lectures, reviewing concepts and tools that are useful for the mini project, but most lab class time will be used for guided discussion and development, student presentations, and some team meetings. However, some lab classes may be used to discuss and solve subject related lab assignments given by the respective teachers of courses.

Generally, students will be taking up mini projects individually. However, in situations when they are working in teams, the individual responsibilities should be planned and documented throughout the phases of the project.

Students are expected to choose an appropriate project topic in consultation with the teacher, and do a short presentation that "pitches" the idea to the teacher and the class. While there is some flexibility in project selection, students should keep in mind the "capstone" nature of this class. Students must develop projects that demonstrate that they have a working knowledge of basic and advanced concepts in computer science and also demonstrate a reasonable knowledge of recent developments in computer science. Each project should include non-trivial software development that has been approved by the teacher/instructor.

With an approved project, students will proceed through a standard sequence of software development stages, beginning with a requirements analysis and specification, and concluding with a final evaluation. A complete detail of the all project stages is given in the milestones section as well as summarized in "TOPICAL CALENDAR" section. At the end of each stage, each individual/team must produce a written report giving stage-specific documentation and describing the work performed, problems encountered, and decisions made. For team projects, the report must include a meeting log and breakdown of tasks by team member. One week before the completion of each stage, there will be a presentation from each project that previews the progress and results in that phase, for in-class discussion and suggestions for refinement in the following week. For these intermediate stage presentations, team members will rotate through as "presenter" for the team, and each student must make at least two intermediate-stage presentations (for a 3-person team this means that there will have to be multiple presentations on the same stage).

In the case of a group project, each member of the group must present the entire project, highlighting their individual contributions toward the project's success, and a short summary of each individual's contributions should be included in the final report as well.

#### SUBMISSION OF DELIVERABLES

Final project report, including all the deliverables, is required to be submitted strictly as per notified schedule.

#### **EVALUATION AND GRADING**

Students work on a single project throughout the duration of this course, and their course grade is calculated based on the grades for individual aspects and milestones. The project will be graded for completeness, content, correctness, quality of presentation (oral and written reports), team work (in case of group project), and the demonstration of the student's knowledge in the computer science field.

As per the University norms Mini Project Report shall be finally evaluated by the external examiner at the end of the semester. However, there will be continuous monitoring of the progress and evaluation of the Mini Project during the semester and the distribution of marks shall be as follows:

Proposal	5 %
Presentations 1 & 2	
Presentations-1: Proposal/Synopsis	15 %
Presentations-II: SRS & Design	
Progress Report 1  Requirements/Specification and Planning and Analysis	20 %
Progress Report 2 System/Research Design	10 %
Progress Report 3	10 %

#### **Implementation and Testing**

#### **Final Deliverables:**

Final Presentation 10 %

Technical Report (including final source code) 30 %

## REQUIRED TEXTS/READING/REFERENCES

Readings and references are project-specific, and will be determined by students/project groups, with approval of the teacher. All the resources used should be properly referenced.

Students will be making extensive use of external references for their project, and should be vigilant in maintaining high standards with regard to attribution and avoidance of plagiarism. If there are questions about how to deal with any such matters, the student should discuss the matter with the teacher concerned to make sure there are no misunderstandings.

### ATTENDANCE POLICY

Attendance is vital for this class, since discussions, regular oral presentations and progress reports will have a strong impact on the ability to complete the project. You may be dropped from the course for missing more than two consecutive scheduled meetings/presentations.

#### **LATE POLICY**

Late work will not be accepted. In case of any unavoidable situations, make requests with the teacher/instructor to reschedule the assigned work/task on case to case basis, if possible.

## MILESTONES FOR MINI-PROJECT

#### 1. Deciding and Registering the topic/title of the mini-project

All the students are required to decide the topic/title for *real life software project*, which they want to design, develop and implement. In finalizing the proposed work topic, they may take help from concerned teachers/mentor in the lab. Decided topic/title *needs to be approved* by the concerned teacher/mentor in the lab.

**Parallel Activity:** Keep preparing the brief summary (synopsis/proposal) of the proposed work as per the given format.

# 2. Submission of brief summary (synopsis/proposal) of the proposed work (As Per the Prescribed Format)

Once the project topic/title is decided and approved, all students are required to **submit** and **present** a brief summary of the proposed work (synopsis/proposal), clearly specifying the client's requirements for which the application software is being developed and the main features of the proposed software. After incorporating the suggestions of the teachers, if any, the final version of the summary of the proposed work (synopsis/proposal) should be submitted to concerned teacher in the lab.

## 3. Submission Requirement specifications, planning and analysis

After the submission and presentation of summary of the proposed work (synopsis/proposal), the students need to submit progress report-1, which includes: analysis modeling and related diagrams (please refer to the Topical calendar for more insights).

#### 4. System Analysis and Design Phase (Submission of SRS document)

All students are required to study and analyze the present system (or existing manual system or proposed system), and all the findings should be **submitted** and **presented** in the form of *SRS document* along with *gantt chart* 

(using appropriate gantt tool, like GanttProject). While doing so, they may actively be involved with client, and/or teachers/Mentors for discussion. Some of the templates/formats of the typical *SRS document* are being attached for your reference.

#### Refer for SRS template:

- http://krazytech.com/projects/sample-software-requirementsspecificationsrs-report-airline-database
- www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc

Discuss the *SRS document* with the concerned teacher/Mentor in the lab, incorporate suggestions (if any), and maintain the different versions of SRS document. Students are required to present and submit *the final signed version* of *SRS document* to the concerned teacher/Mentor in the lab.

*SRS document* should contain the ER diagram, Data Flow Diagram and Data Dictionary. You should also prepare important UML diagrams like Use Case diagrams, Activity diagrams, class diagrams, behaviour model and/or state transition diagram etc.

Students are advised to use standard tools for drawing UML diagrams, DFD, ER Diagrams, etc. Examples of some typical tools are listed below:

- UML diagrams using automated tool such as StarUML, BOUML etc.
- Data Flow Diagram (DFD) with different levels using tools such as Lucidchart, Visual Paradigm, etc.
- E-R Diagram with the help of automated tools such as ERDPlus, Smartdraw, etc.

**Parallel Activity:** In the mean time, you may learn and practice the tools necessary to develop the proposed software, and finalize the detailed database design, including populated tables. Also students should start the coding in parallel with their presentation of **SRS document**.

#### 5. System Development (Coding/Testing)

Start the development of the system as per the design specifications discussed in *SRS document*. Coding should be well documented. *Technical Report* specifying the brief technical specifications and documenting the working of each major modules/methods should be submitted. Students are required to properly maintain the following during the system development:

• A clear design of working database of the system using a popular DBMS such as ORACLE, SQL Server, MYSQL, etc.

#### Deployment/Implementation

Deploy/Implement the developed system on the client site (actual user site) and *obtain* user acceptance letter, specifying that the developed system is working satisfactorily and is as per the specified requirements. Better you prepare an installation copy for your software along with installation manual.

**Parallel Activity:** Keep writing and preparing the **Final project Report** as per the standard format. (**Refer:** Format for Project Report.pdf)

#### 6. Final Evaluation

Demonstrate the working of system to the audience (teacher, Mentor students, clients), specifying the design and main features of the developed system. The **final project report** and an **oral presentation** should be submitted as per standard project report format/template. The user manual must be a part of the final project report and should be written in explanatory manner so that anyone can operate the system using this manual. Hardcopy as well as softcopy of the all the reports (like SRS Document, Technical report, Final Project report) should be submitted to the concerned teacher/Mentor in the lab. Softcopy of complete project (code), database and necessary files (preferably, installation software, along with installation manual) should also be submitted (**Refer:** Format for Project Report.pdf).

TOPICAL CALENDAR			
S.No.	Project Stage/Activities	Deliverables	Duration
01	Deciding and Registering the Topics/Titles of the Mini project	Registration of the Project topic/title Parallel Activity: Synopsis preparation	1-2 Week
Brief Summary		Submission of Preliminary Proposal/Synopsis	2-3 Week
02	(Synopsis/Proposal) of the proposed work	Presentation 1 (Proposal/Synopsis) Parallel Activity: Requirement Specifications and Planning and Analysis	3-4 Week
03	Requirement Specification, Planning and Analysis	Progress Report 1 Submission	5-7 Week
		Progress Report 1 Submission (SRS & Design Document)	7-8 Week
04	System/Research Design	Presentation 2 (SRS & Design) Parallel Activity: Implementation, Deployment and Testing	8-9 Week
05	Implementation, Deployment and Testing	Progress Report 2 Submission (Technical Report)  Includes brief code walkthrough Source code Test results and discussion	9-12 Week
	Evaluation and Refinement	Presentation 3 (Final presentation) Parallel Activity: Evaluation and Refinement (Final Report)	13-14 Week
06		Final Report submission	14 Week

Note: The last date of submission of various activities will be notified separately from time to time.

# LAB ASSIGNMENTS

# **APPENDIX-I**

# Template for the Index of Lab File

WEEK NO.		PROBLEMS WITH DESCRIPTION	PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
	1#			
1	2#			
	3#			
	1#			
2	2#			
	3#			
	1#			
3	2#			
	3#			

Note: The students should use Header and Footer mentioning their roll no. & name in footer and page no in header.

- 1. Write a program to find product of two user supplied integers and if the product is equal to or lower than 5000 then return sum of the two numbers.
- 2. Write a program to print sum of first 10 numbers.
- 3. Write a program to Iterate the supplied list of 20 numbers by the user and print only those numbers which are divisible by 5.
- 4. Write a program to check if the given number is a palindrome number or not.
- 5. Write a program to calculate the cube of all numbers from 1 to a given number.

- 1. Write a Program to extract each digit from an integer in the reverse order.
- 2. Write a program to count the total number of digits in a number using a while loop.
- 3. Write a program to display all prime numbers within a range.
- 4. Write a program to use the loop to find the factorial of a given number.
- 5. Write a program to find the sum of digits of a supplied integer.

1. Write a program to print the following pattern using the for loop:

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
```

2. Write a program to print the following start pattern using the for loop:

- 3. Write a program to print characters from a string which are present at an even index numbers.
- 4. Write a program to accept a string from the user and display characters that are present at an even index number.
- 5. Write a program to remove characters from a string starting from n to last and return a new string. *Example: remove\_chars("aligarh", 3) so output must be al.*

- 1. Write a program to create function cal\_sum\_sub() such that it can accept two variables and calculate addition and subtraction. Also, it must return both addition and subtraction in a single return call.
- 2. Write a function to return True if the first and last number of a given list is same. If numbers are different then return False.
- 3. Given a list of numbers. Write a program to turn every item of a list into its square.
- 4. Given a two Python list. Write a program to iterate both lists simultaneously and display items from list 1 in original order and items from list 2 in reverse order.
- 5. Write a program to count the number of occurrences of item 50 from below tuple tp1:

$$tp1 = (50, 10, 60, 70, 50)$$

- 1. Write a program to generate 6 digit random secure OTP.
- 2. Write a program to pick a random character from a given String supplied by the user.
- 3. Write a program to generate a random Password which meets the following conditions
  - a. Password length must be 10 characters long.
  - b. It must contain at least 2 upper case letters, 1 digit, and 1 special symbol.
- 4. Given a two list of numbers, write a program to create a new list such that the new list should contain odd numbers from the first list and even numbers from the second list.
- 5. Write a program to create a numpy array and return array of odd rows and even columns from the numpy array.
- 6. Write a program create a numpy array and sort as per below cases:
  - a. Case 1: Sort array by the second row.
  - b. Case 2: Sort the array by the second column.

- 1. Write a Python program that inputs two tuples and creates a third that contains all elements of the first followed by all elements of the second. (You may use other data types such as lists etc. to make your program work)
- 2. Write a Python program to create a Python dictionary (the dictionary type). Add three names and their phone numbers in the dictionary. The names should act as a keys and phones as their values. Then ask the user a name and print its corresponding phone number.
- 3. Write a Python program having a void function that receives a 4-digit number and calculates the sum of squares of first 2 digits' number and last two digits' number, e.g. if 1233 is passed as argument then function should calculate 12<sup>2</sup> + 33<sup>2</sup>.
- 4. Write a program that inputs a main string and then creates an encrypted string by embedding a short symbol-based string after each character. The program should also be able to produce the decrypted string from encrypted string.
- 5. Write a program to get roll numbers, names and marks of the students of a class (get from user) and store these details in a file called "Marks.data"
- 6. Write a program to accept a string and display the following:
  - a. Number of uppercase characters
  - b. Numbers of lowercase characters
  - c. Total number of alphabets
  - d. Number of digits