COURSE PLAN

Course Code: 18UCSC601		Course Name : Object Oriented Modeling & Design	
Course Teacher : Dr. U.P.Kulkarni		Semester Duration: 17th March to 4th June 2022	
Semester: 6	Division: A	Credits :4 / Hours: 52	Type : 4-0-0

Coverage: As specified in the PEM based syllabus and Academic calendar published by the Institution.

Unit	Start Date	End Date	No. of Hours
1	17-3-2022	01-4-2022	10
2	05-4-2022	28-4-2022	12
3	29-4-2022	17-5-2022	10
4	18-5-2022	09-6-2022	10
5	10-6-2022	28-6-2022	10
		TOTAL	52

Assessment Tools :

1. Three assessment tool of written type examination (IAs), each of 20 marks.
2. Course teacher's assessment (CTA) is for 10 marks based on implementation based assignments/ Course project specified by the course teacher/ PEM based syllabus.
3. Preparatory test (Open Book) for each IA will be conducted a day before the IA. It is mandatory for all students to attend this to become eligible to appear for corresponding IA.

Class Test	Date and Time
1	22-4-2022
2	31-5-2022
3	28-6-2022

4. Solutions for every test will be discussed in the class immediately following the test.
5. CIE out of 50 is calculated based on Sum of the best two IAs plus CTA.
6. Semester End Examination (SEE) is of written type for 100 marks, reduced to 50.
7. Final grading is based on sum of CIE (50) and reduced SEE (50).
8. Minimum marks to be scored in CIE are 20 to appear for SEE.
9. Minimum passing marks is 40 out of 100 in SEE.

Class Coverage and Assignments for CTA:

Simple application programs (based on the theory concepts covered in the class) to be written by an individual students in the laboratory course attached with this. Based on the understanding of the concepts, the following course project is to be done in a group of TWO students and collaboration with senior students (6th semester- OOMD course of same division). Course project is considered for CTA assignments of 10 marks.

Class Coverage Sequence through programming:

- 6/8 Hours of common session on Object Oriented Philosophy by course teachers to students of A and B divisions.
  (First and Second day of the semester in the afternoon. 2pm to 5pm)
- Basic concepts of languages like: data types, control structures arrays, strings and EXCEPTIONS will be managed as a part of every class room programming sessions listed below rather than separate sessions.

Class room Programming Sessions

1. Java Program to implement abstract data type (ADT) for the given scenario.
   Learning Objectives: [CO-1]
   a) To understand the principles of Object Oriented Programming Paradigm-OOP: Managing complexity.
   b) To Understanding class path setting.
   c) To know how to compile and run java code at command prompt.
   d) To know class structure, methods, package and Access specifiers.
2. Java Program to understand abstract class, Interface, Inheritance, Composite Objects for the given scenario.
   Learning Objectives: [CO-1, 2, 4]
   a) To understand the principles of Object Oriented Programming

   [illegible] be enforced in the design

   [illegible] specifier w.r.t packages and inheritance

   Learning Objectives: [CO-1, 2, 5]
   a) To know access specifier- default, private, protected and public
   b) To know modularity concepts

5. *SELF LEARNING EXERCISE: Java Program to understand use of utility classes.*
   Note: Student group will be asked to give demonstration.
   Learning Objectives: [CO-1, 3]
   a) Quick product development, Use of standard utilities like: Stack and List.
6. Java programs to demonstrate thread concepts, thread features and its application in operating system to know conflicts and solution.
   Learning Objectives: [CO-1, 7]
   a) Quick product development, Use of standard utilities like: Stack and List.

Course project [CO-7]
// In collaboration with senior students of OOMD course of respective semester.

Develop a GUI based system to provide a solution to a multiple producers and consumers of sensor data.

Preprocessed data set is made available in a multiple files.

Here are the specific **requirements** of the system to be implemented.

R1. Producer threads (multiple) should read a file name and then its content (in the predefined format) and make it available to a common place called data buffer.

R2. Producer waits until there is space in the Data Buffer of fixed size.

R3. Producer starts reading continuously the file from the beginning when it reaches to the end of the file.

R4. Data buffer size is to be read when the system starts. It should be less or equal to ten.

R5. Number of producers and consumers to be read at the start of the system. They should be always even numbered and less than or equal to 10.

R6. Multiple consumers read records from the Data Buffer and write them in to a file.

R7. Every consumer have their specific output file whose name begin with <consumer thread ID>_< Time stamp>.data.

R8. Consumer waits until there is a record in the Data Buffer to read.

R9.Once a producer program starts reading a file; no other producer programs should be permitted to read the same file.

R10. Input data sets consist of the record: Sensor ID: Parameter Name and Value separated by # stored in multiple file.

Example for data set: SID-007: temp#45 indicating that sensor ID is 7 producing a temperature data value of 45 degree centigrade

All invalid records read by all producers to be stored in a separate single file called errorLOG.data in the format : <Thread ID> : <record read > : < error details>

Note:

1. Any other **suitable assumptions** can be made in visualizing the scope and specification details of the problem statement with real life domain mapping and for missing, ambiguous specification, inconsistent requirements or incorrect statements specified.

2. **Use of** exceptions, UI, Threads, Streams/ Files support of java language is to be focused during implementation.

3. **Submission deadline:**
   a) Synopsis- 23rd March 2022.
   b) Final report submission- within a week of completion of IA-2 in Google class room.
   c) Demonstration through group presentation immediately after 2nd IA.

Basic System Architecture: < for three producers and three consumer and Data Buffer of size=7 record instance>

Report is to be prepared focusing on Software Engineering principles as listed below:

1. Preparation of detailed requirement specification – **5 Marks**
2. Preparation of Class and associated UML Diagram- **10 Marks**
3. Preparation of UI Design- **5 Marks**
4. Preparation of Test Case Design- **20 Marks**
5. Emphasis on coding style/ practices, writing robust code, Testable code and documentation and report writing- **20 Marks**
6. Preparation of presentation slides- **10 Marks**
7. Demonstration of the project work- **30 Marks.**

Total 100 marks reduced to 10 for CTA (theory) and CTA of laboratory is evaluated based on the project demonstration in group for 50 marks.

Learning Objectives:

To Know:

1. Project management activities.
2. Preparation of SRS.
3. The use of UML tools for design.
4. How to visualize system architecture.
5. How to prepare test cases.
6. Good coding practices.
7. Professional report writing.
8. The use of tools.
9. To work in a group and enable self-learning from pears.

7. Java program to read from and write in to the files.
   Learning Objectives: [CO-1, 8]
   a) To know storage and retrieval operation required for every applications.

8. Java program to build simple GUI.
   Learning Objectives: [CO-1, 9]
   a) To know use of AWT/Swing features.
   b) To know event handling features of java language.

Reference Books:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.
2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Course Plan

Course Teachers: Sharada H N (B division)	Course Code: 18UCSC400 – ARM Processor	3-0-0-0 : 3	39 Hrs	4th Semester, B div	17-03-2022 to 04-07-2022

Activity/ Lesson	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14	W 15	W 16	W 17	W18 –W22
Chapter – 1 ARM Embedded Systems and Processor Fundamentals	3	-	-	-	-	-	IA 1	-	-	-	-	-	IA 2	-	-	-	IA 3	Lab Exam & Semester End Exams Week
	-	3 + 1	-	-	-	-		-	-	-	-	-		-	-	-		
	-	-	3 + 1	-	-	-		-	-	-	-	-		-	-	-		
	-	-	-	1	-	-		-	-	-	-	-		-	-	-		
Chapter-2 ARM Instruction Set	-	-	-	3	-	-		-	-	-	-	-		-	-	-		
	-	-	-	-	3 + 1	-		-	-	-	-	-		-	-	-		
	-	-	-	-		3 +1		-	-	-	-	-		-	-	-		
	-	-	-	-	-	-		-	-	-	-	-		-	-	-		
Chapter-3 Introduction to THUMB and ARM programming	-	-	-	-	-	1		-	-	-	-	-		-	-	-		
	-	-	-	-	-	-		3+1	-	-	-	-		-	-	-		
	-	-	-	-	-	-	-	-	3+1	-	-	-		-	-	-		
	-	-	-	-	-	-	-	-	-	3+1	-	-		-	-	-		
Chapter-4 Exceptions and Interrupt handling schemes	-	-	-	-	-	-	-	-	-	-	3+1	-		-	-	-		
	-	-	-	-	-	-	-	-	-	-	-	3+1		-	-	-		
	-	-	-	-	-	-	-	-	-	-	-	1		3	-	-		
Chapter-5 LPC2148 ARM CPU Peripherals- GPIO, PLL & Timers	-	-	-	-	-	-	-	-	-	-	-	-		1	-	-		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3+1	3+1		

[SHARADA HN]

IA	Topics	No. Of Hours
IA-I	Chapter- 1, Chapter-2	16 Hrs
IA-II	Chapter- 3, Chapter-4 (up to Interrupts)	15 Hrs
IA-III	Chapter -5	08 Hrs

Note:

1. Number in the 'cell' indicates the number of hours/class used in that week to cover that chapter; 'W' indicates week number of the semester.
2. The entry 1 in the 2nd week (W2) is just indicative, but there will be a class test to assess the capability of the students for deciding the pedagogy.
3. Final grade (absolute) is calculated based on Semester End Examination (SEE) marks (100 marks reduced to 50) + Continuous Internal Evaluation (CIE) marks (50).
4. CIE marks (50) is based on sum of the best two Internal Assessment (IAs) (40 Marks) plus Course Teacher's Assessment (CTA) marks (10 marks).
5. CTA (10 Marks): The details are shown in Table 1.
6. 85 % attendance is mandatory.

Action Plan based on observation taken from previous Attainment of Outcomes:

Programming assignment as part of the CTA and Course assignments are planned to enhance the following abilities:

1. Understand the internal architecture, instruction set of ARM7 microcontroller.
2. Understand the use of interrupts related to ARM 7.
3. Write APL / C program for a given real time application.
4. Basic programming and methods used to interface ARM 7 to devices such as motors, LCD, ADC, DAC etc.

Activity	Date of conduction	Description	Course Outcome	Marks
1	23-04-2022	Class test/Quiz as a preparatory for IA 1	CO 1	4
2	28-05-2022	Class test/Quiz as a preparatory for IA 2	CO 2, CO3	4
3	25-06-2022	Programming Assignment	CO 4, CO 5	2

Table 1: LIST OF ACTIVITIES / ASSIGNMENTS

Learning Resources:

1. Andrew N. Sloss, ARM System Developer's guide, ELSEVIER Publications, 2016
2. William Hohl, ARM Assembly Language, CRC Press.
3. Steve Furber, ARM System-on-chip Architecture by, Pearson Education, 2012
4. LPC 2148 USER MANUAL.

Course Plan (Academic Year 2021-22, Evn Semester)

Course Teacher	Course																	
Vidyagouri B Hemadri	18UCSC401 FAFL							3-0-0 : 3		39 Hrs		4th Sem UG CSE					17-03-2022 to 05-07-2022	

Activity/ Lesson	Assessment tool	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18-19
Unit – 1 Introduction to Finite Automata (9 Hrs IA-I)	Pre Test Quiz involving GATE question Written assignment OR Presentation	3	3	3				IA-1					IA-2					IA-3	Lab Exam and Semester End Exam as per the Time table given by COE's Office
Unit – 2 Regular Expressions and Languages (7 Hrs IA-I & IA-II)	Quiz involving GATE question Written assignment OR Presentation				3	3	1+2 Revision												
Unit – 3 Context-Free Grammars and Languages (5 Hrs IA-II) Compiler Writing Tools (3 Hrs IA-II)	Quiz involving GATE question Written assignment OR Presentation Write simple LEX and YACC programs to recognize the patterns and expressions.								3	3	2								

Course Teacher Assessment (CTA) – 10 Marks

This is based on the following components

1. Quiz → 5 Marks
2. Written assignment covering entire syllabus → 5 Marks

OR

2. Presentation → 5 Marks
   i. Use of regular expression in various applications
   ii. Use of finite automata in various applications
   iii. Use of context free grammar in various applications

Reference Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman Introduction to Automata Theory, Languages and Computation, Pearson Education, 3/E, 2013.
2. Peter Linz, An Introduction to Formal Languages and Automata, Narosa Publishing House, 5/E, 2011.
3. John R. Levine and Tony Mason and Doug Brown UNIX programming tools 2/E. 1992.
4. John Martin, Introduction to languages and theory of computation, Tata McGraw-Hill, 4/E, 2010.

Course Teacher

Vidyagouri B Hemadri

Email: vidya.gouri@gmail.com

9448776104

COURSE PLAN

Course Code: 18UCSC402		Course Name : Object Oriented Programming	
Course Teacher : Dr. U.P.Kulkarni		Semester Duration: 17th March to 4th June 2022	
Semester: 4	Division: A	Credits :4 / Hours: 52	Type : 4-0-0

Coverage: As specified in the PEM based syllabus and Academic calendar published by the Institution.

Unit	Start Date	End Date	No. of Hours
1	17-3-2022	31-3-2022	09
2	02-4-2022	16-4-2022	09
3	18-4-2022	12-5-2022	12
4	14-5-2022	04-6-2022	10
5	06-6-2022	25-6-2022	12
		TOTAL	52

Assessment Tools :

1. Three assessment tool of written type examination (IAs), each of 20 marks.
2. Course teacher's assessment (CTA) is for 10 marks based on implementation based assignments/ Course project specified by the course teacher/ PEM based syllabus.
3. Preparatory test (Open Book) for each IA will be conducted a day before the IA. It is mandatory for all students to attend this to become eligible to appear for corresponding IA.

Class Test	Date and Time
1	23-4-2022
2	30-5-2022
3	27-6-2022

4. Solutions for every test will be discussed in the class immediately following the test.
5. CIE out of 50 is calculated based on Sum of the best two IAs plus CTA.
6. Semester End Examination (SEE) is of written type for 100 marks, reduced to 50.
7. Final grading is based on sum of CIE (50) and reduced SEE (50).
8. Minimum marks to be scored in CIE are 20 to appear for SEE.
9. Minimum passing marks is 40 out of 100 in SEE.

16/3/22

Assignments for CTA:

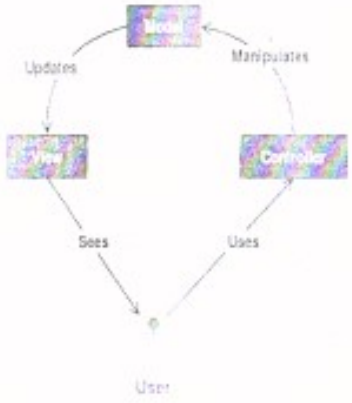
- Following assignments are to be done in a group of two students.
- This group has to collaborate with 4th Semester A division students to mentor their course project.
- Evaluation will be done through seminar and report submitted. Evaluation starts immediately after 2nd IA.

Course work description:

1. Prepare a business scenario and apply **Model View Controller** (MVC) design pattern. Implement the pattern in appropriate language.

   The **Model View Controller** (MVC) design pattern specifies that an application consist of a data model, presentation information, and control information. The pattern requires that each of these be separated into different objects.

   MVC is more of an architectural pattern, but not for complete application. MVC mostly relates to the UI / interaction layer of an application. You're still going to need business logic layer, maybe some service layer and data access layer.



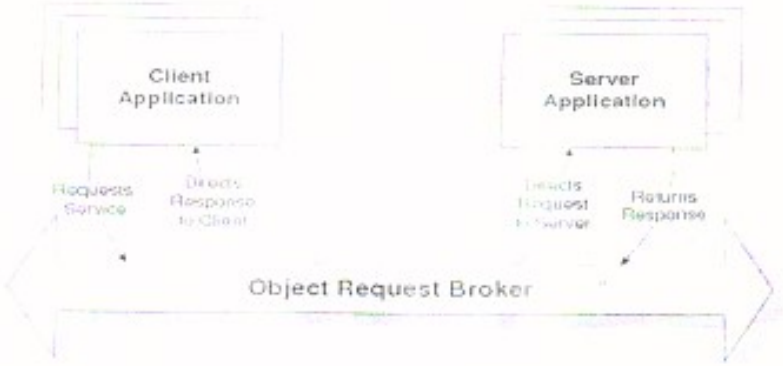
Design components

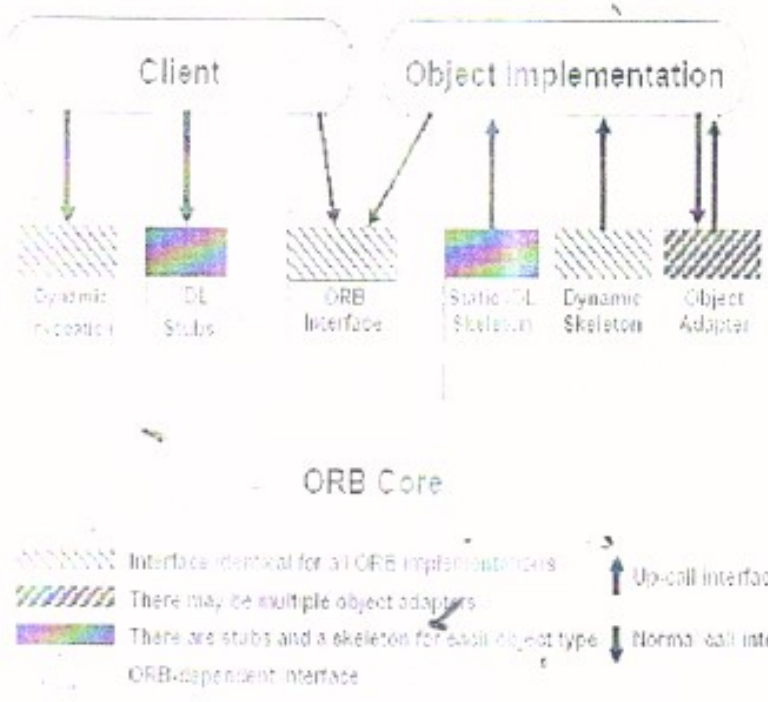
- The **Model** contains only the pure application data; it contains no logic describing how to present the data to a user.
- The **View** presents the model's data to the user. The view knows how to access the model's data, but it does not know what this data means or what the user can do to manipulate it.
- The **Controller** exists between the view and the model. It listens to events triggered by the view (or another external source) and executes the appropriate reaction to these events. In most cases, the reaction is to call a

method on the model. Since the view and the model are connected through a notification mechanism, the result of this action is then automatically reflected in the view.

2. Prepare the business scenario and write a program to show the **CORBA**. The Common Object Request Broker Architecture (CORBA) is a standard defined by the Object Management Group (OMG) that enables software components written in multiple computer languages and running on multiple computers to work together.

   Note: As a preparatory work, you are expected to implement **Java RMI**.





3. Presentation of **Reusable objects and writing test scripts** work completed in Software Engineering course of 5th semester.

4. **Mentoring** the course project of Object Oriented Programming of 4th semester A division.

Reference Books:

1) Michael Blaha, James Rumbaugh, "Object-Oriented Modeling and Design with UML", 2 E. Pearson Education, 2007.
2) Frank Buschmann, RegineMeunier, Hans Rohnert, Peter Sommerlad, Michael Stal, "Pattern-Oriented Software Architecture", A System of Patterns Volume 1, John Wiley and Sons, 2006.
3) Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice", 2 E. Pearson Education, 2003.
4) Grady Boochetai, "Object-Oriented Analysis and Design with Applications", 3 E. Pearson Education, 2007.
5) Ali Bahrami, "Object oriented systems development", McGrawHill, 1999.
6) Mary Shaw and David Garlan, "Software Architecture Perspectives on an Emerging Discipline", Prentice-Hall of India, 2007.