FR. Conceicao Rodrigues College of Engineering

Department of Computer Engineering

T.E. (Computer) (semester V)

(2022-2023)

Subject: Software Engineering

Subject Code: CSC502

Course Outcomes and Assessment Plan

Course Code:	Course Title	Credit
CSC502	Software Engineering	3

Prerequisite: Object Oriented Programming with Java , Python Programming	
Course Objectives:	
1	To provide the knowledge of software engineering discipline.
2	To apply analysis, design and testing principles to software project development.
3	To demonstrate and evaluate real world software projects.
Course Outcomes: On successful completion of course, learners will be able to:	
1	Identify requirements & assess the process models.
2	Plan, schedule and track the progress of the projects.
3	Design the software projects.
4	Do testing of software project.
5	Identify risks, manage the change to assure quality in software projects.

Module		Content	Hrs
1		**Introduction To Software Engineering and Process Models**	7
	1.1	Software Engineering-process framework, the Capability Maturity Model (CMM), Advanced Trends in Software Engineering	
	1.2	Prescriptive Process Models: The Waterfall, Incremental Process Models, Evolutionary Process Models: RAD & Spiral	
	1.3	Agile process model: Extreme Programming (XP), Scrum, Kanban	
2		**Software Requirements Analysis and Modeling**	4
	2.1	Requirement Engineering, Requirement Modeling, Data flow diagram, Scenario based model	
	2.2	Software Requirement Specification document format(IEEE)	
3		**Software Estimation Metrics**	7
	3.1	Software Metrics, Software Project Estimation (LOC, FP, COCOMO II)	
	3.2	Project Scheduling & Tracking	
4		**Software Design**	7
	4.1	Design Principles & Concepts	
	4.2	Effective Modular Design, Cohesion and Coupling, Architectural design	
5		**Software Testing**	7
	5.1	Unit testing, Integration testing,Validation testing, System testing	
	5.2	Testing Techniques, white-box testing: Basis path, Control structure testing black-box testing: Graph based, Equivalence, Boundary Value	
	5.3	Types of Software Maintenance, Re-Engineering, Reverse Engineering	
6		**Software Configuration Management, Quality Assurance and Maintenance**	7

	6.1	Risk Analysis & Management: Risk Mitigation, Monitoring and Management Plan (RMMM).	
	6.2	Quality Concepts and Software Quality assurance Metrics, Formal Technical Reviews, Software Reliability	
	6.3	The Software Configuration Management (SCM), Version Control and Change Control	
			39

Textbooks:	
1	Roger Pressman, “***Software Engineering: A Practitioner‘s Approach”***, 9th edition , McGraw-Hill Publications, 2019
2	Ian Sommerville, ***“Software Engineering”***, 9th edition, Pearson Education, 2011
3	Ali Behfrooz and Fredeick J. Hudson, ***"Software Engineering Fundamentals"***, OxfordUniversity Press, 1997
4	Grady Booch, James Rambaugh, Ivar Jacobson, ***“The unified modeling language user guide”***, 2nd edition, Pearson Education, 2005
References:	
1	Pankaj Jalote, ***"An integrated approach to Software Engineering"***, 3rd edition, Springer, 2005
2	Rajib Mall, ***"Fundamentals of Software Engineering"***, 5th edition, Prentice Hall India, 2014
3	Jibitesh Mishra and Ashok Mohanty, ***“Software Engineering”***, Pearson , 2011
4	Ugrasen Suman, ***“Software Engineering – Concepts and Practices”***, Cengage Learning, 2013
5	Waman S Jawadekar, ***“Software Engineering principles and practice”***, McGraw Hill Education, 2004

Useful Links	
1	https://nptel.ac.in/courses/106/105/106105182/
2	https://onlinecourses.nptel.ac.in/noc19_cs69/preview
3	https://www.mooc-list.com/course/software-engineering-introduction-edx

Course Outcomes:

Upon completion of this course students will be able to:

CSC502.1: Identify requirements and assess the process models. (Analyze)
CSC502.2: Plan, schedule and track the progress of the projects. (Apply)
CSC502.3: Design the software projects. (Apply)
CSC502.4: Do testing of software project. (Apply)
CSC502.5: Identify risks, manage the change to assure quality in software projects. (Analyze)

Mapping of CO and PO/PSO

Relationship of course outcomes with program outcomes: Indicate 1 (low importance), 2 (Moderate Importance) or 3 (High Importance) in respective mapping cell.

	PO1 (Eng Knwlg)	PO2 (Ana)	PO3 (Design)	PO4 (investiga)	PO5 (tools)	PO6 (engg Soci)	PO7 (Env)	PO8 (Eth)	PO9 (ind Team)	PO10 (comm.)	PO11 (PM)	PO12 (life Long)
CSC502.1	1	1	3	3						1		
CSC502.2	1	3							3	3	3	2
CSC502.3	1	1	3						3			
CSC502.4				2	2							
CSC 502.5	1	2										
TOTAL	**4**	**7**	**6**	**5**	**2**				**6**	**4**	**3**	**2**
CO-PO Matrix	**1**	**2**	**3**	**3**	**2**				**3**	**2**	**3**	**2**

CO	**PSO1**	**PSO2**
CSC502.1	3	3
CSC502.2	3	3
CSC502.3	3	3
CSC502.4	3	3
CSC502.5	3	3
Course to PSO	2.38	2.384

Course Outcome	**Competency**	**Performance Indicator**

CSC502.1	1.4 Demonstrate competence in specialized engineering knowledge to the program	1.4.1 Apply theory and principles of Computer Science and engineering to solve an engineering problem
	2.2 Demonstrate an ability to formulate a solution plan and methodology for an engineering problem	2.2.4 Compare and contrast alternative solution/methods to select the best methods
	3.1 Demonstrate an ability to define a complex/ open-ended problem in engineering terms	3.1.1 Able to define a precise problem statement with objectives and scope. 3.1.2 Able to identify and document system requirements from stake- holders. 3.1.3 Able to review state-of-the-art literature to synthesize system requirements. 3.1.4 Able to choose appropriate quality attributes as defined by ISO/IEC/IEEE standard. 3.1.5 Explore and synthesize system requirements from larger social and professional concerns. 3.1.6 Able to develop software requirement specifications (SRS).
	4.3 Demonstrate an ability to analyze data and reach a valid conclusion	4.3.1 Use appropriate procedures, tools and techniques to and analyze collect data 4.3.2 Critically analyze data for trends and correlations, stating possible errors and limitations 4.3.4 Synthesize information and knowledge about the problem from the raw data to reach appropriate conclusions.
	10.2 Demonstrate competence in listening, speaking, and presentation	10.2.1 Listen to and comprehend information, instructions, and viewpoints of others

CSC502.2	1.4 Demonstrate competence in specialized engineering knowledge to the program	1.4.1 Apply theory and principles of Computer Science and engineering to solve an engineering problem
	2.2 Demonstrate an ability to formulate a solution plan and methodology for an engineering problem	2.2.2 Identify functionalities and computing resources. 2.2.3 Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions. 2.2.4 Compare and contrast alternative solution/methods to select the best methods 2.2.5 Compare and contrast alternative solution processes to select the best process.
	2.4 Demonstrate an ability to execute a solution process and analyze results	2.4.2 Analyze and interpret the results using contemporary tools. 2.4.3 Identify the limitations of the solution and sources/causes. 2.4.4 Arrive at conclusions with respect to the objectives.
CSC502.3	1.4 Demonstrate competence in specialized engineering knowledge to the program	1.4.1 Apply theory and principles of Computer Science and engineering to solve an engineering problem
	3.3 Demonstrate an ability to select optimal design scheme for further development	3.3.2 Consult with domain experts and stakeholders to select candidate engineering design solution for further development
	3.4 Demonstrate an ability to advance an engineering design to defined end state	3.4.1 Able to refine architecture design into a detailed design within the existing constraints. 3.4.2 Able to implement and integrate the modules. 3.4.3 Able to verify the functionalities and validate the design.
	9.2 Demonstrate effective individual and team operations– communication, problem-solving, conflict resolution and leadership skills	9.2.1 Demonstrate effective communication, problem-solving, conflict resolution and leadership skills 9.2.2 Treat other team members respectfully 9.2.3 Listen to other members 9.2.4 Maintain composure in difficult situations
	9.3 Demonstrate success in a team-	

	based project	9.3.1 Present results as a team, with smooth integration of contributions from all individual efforts
CSC502.4	4.1 Demonstrate an ability to conduct investigations of technical issues consistent with their level of knowledge and understanding	4.1.2 Able to choose appropriate procedure/algorithm, dataset and **test cases.** 4.1.3 Able to choose appropriate hardware/software tools to conduct the experiment. **(testing)**
	5.1 Demonstrate an ability to identify/create modern engineering tools, techniques and resources	5.1.1 Identify modern engineering tools, techniques and resources for engineering activities
	5.2 Demonstrate an ability to select and apply discipline-specific tools, techniques and resources	5.2.2 Demonstrate proficiency in using discipline-specific tools
CSC502.5	1.4 Demonstrate competence in specialized engineering knowledge to the program	1.4.1 Apply theory and principles of Computer Science and engineering to solve an engineering problem
	2.3 Demonstrate an ability to formulate and interpret a model	2.3.1 Able to apply computer engineering principles to formulate modules of a system with required applicability and performance. 2.3.2 Identify design constraints for required performance criteria.

Assessment Tools:

Course Outcome	**Assessment Tool** **Direct (weightage: 80%)**	**Assessment Tool Indirect (weightage = 20%)**
CSC502.1: Identify requirements and assess the process models. (Analyze)	Test 1 (20%) Lab Assignment 1 and 2 (25%) Assignment 1(20%) Quiz (10%) University Exam (25%)	Course Exit Survey

CSC502.2: Plan, schedule and track the progress of the projects. (Apply)	Test2 (20%) Lab assignment 3 and 4(20%) Assignment 2(20%) Quiz (10%) University Exam (30%)	

CSC502.3: Design the software projects. (Apply)	Test 1 and 2(20%) Lab Assignment 2 and 5(20%) Assignment 1(20%) Quiz (10%) University Exam(30%)	
CSC502.4: Do testing of software project. (Apply)	Test 2(20%) Lab Assignment 6,7 and 8(20%) Assignment 2(20%) Quiz (10%) University Exam(30%)	
CSC502.5: Identify risks, manage the change to assure quality in software projects. (Analyze)	Test 2(20%) Lab Assignment 9 and 10(20%) Assignment 2(20%) Quiz (10%) University Exam(30%)	

CO Assessment Tools:

CSC502.1: Direct Methods(80%): Unit Test+ Lab_Assignmnet+Assignment+Quiz+UniExam

CO2dm = 0.2T +0.25Lab+ 0.2Assig+ 0.1Quiz+0.25Uniexam

Indirect Methods(20%): Course exit survey

CO1idm

CSC502.1 = 0.8*CO1dm + 0.2* CO1idm

CSC502.2:Direct Methods (80%):

Unit Test+Lab_Assignmnet+Assignment2+Quiz+UniExam

CO2dm = 0.2T +0.2Lab+ 0.2Assig+ 0.1Quiz+0.3Uniexam

InDirect Methods(20%): Course exit survey

CO2idm

CSC502.2 = 0.8*CO2dm + 0.2* CO2idm

CSC502.3: Direct Methods (80%):

Unit_Test2+Lab_Assignment+Assignment+Quiz+UniExam

CO3dm =0.20T +0.2Lab_marks+0.2Assig+ 0.1Quiz+ 0.3Uniexam

InDirect Methods(20%): Course exit survey

CO3idm

CSC502.3 = 0.8*CO3dm + 0.2* CO3idm

- -

CSC502.4: **Direct Methods (80%):** Lab assignments+Uniexam+project

CO4dm=**0.20T +0.2Lab_marks+0.2Assig+ 0.1Quiz+ 0.3Uniexam**

InDirect Methods(20%): Course exit survey

CO4idm

CSC502.4 = 0.8*CO4dm + 0.2* CO4idm

CSC502.5:

Direct Methods (80%): Lab assignments+Uniexam+project

CO5dm=**0.20T +0.2Lab_marks+0.2Assig+ 0.1Quiz+ 0.3Uniexam**

InDirect Methods(20%): Course exit survey

CO5idm

CSC502.5= 0.8*CO5dm + 0.2* CO5idm

Rubrics for Assignments:

Performance Indicator	**Excellent**	**Good**	**Below average**
Timeline(2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)
Organization (2)	Well organized, neat and clear handwriting, neat diagrams with all labels (2)	Organized to some extent, diagrams and handwriting is neat with some missing labels (1)	Poorly organized, diagrams incomplete (0.5)
Level of content (3)	All points are covered and answered accurately (3)	Some important points are omitted /Addressed minimally (1-2)	Many important points are missing and the answers are not accurate. (1-0)
Knowledge about the topic (3)	All Concepts of a topic are clear and knows the application to real world problems (3)	All Concepts of a topic are mostly clear lacks understanding about the application to real World problems (2-1)	Poor understanding of concepts and application to real world problems (1-0)

Lesson Plan

Modes of Content Delivery:

i	Class Room Teaching	v	Self Learning Online Resources	ix	Industry Visit
ii	Tutorial	vi	Slides	x	Group Discussion
iii	Remedial Coaching	vii	Simulations/Demonstrations	xi	Seminar
iv	Lab Experiment	viii	Expert Lecture	xii	Case Study

Module 1: **Introduction to Software Engineering and Process Models**

Lecture No.	**Date**		**Topic**	**Content Delivery Method**
	Planned	**Actual**		
1	19/7/2022	19/7/2022	what is software? what is software engineering? its importance, SE as Layered technology	Slides, case study
2	20/7/2022	20/7/2022	SDLC Phases, waterfall model	Slides, Case Study
3	21/7/2022	21/7/2022	Iterative model, Incremental model, prototyping	Slides, Case Study
4	26/7/2022	26/7/2022	Spiral model, RAD model	Slides, Case Study
5	27/7/2022	27/7/2022	Generic process framework activities, umbrella activities, process flow patterns, Capability Maturity Model	Slides
6	28/7/2022	28/7/2022	Agile models basics, extreme programming	Slides, Case Study
7	1/8/2022	1/8/2022	scrum and Kanban Agile frameworks	Slides, Case Study

Module 2: **Introduction to Software Engineering and Process Models**				
Lecture No.	**Date**		**Topic**	**Content Delivery Method**
	Planned	**Actual**		
1	2/8/2022	2/8/2022	Requirement engineering (Tasks)	Slides, Lab experiment
2	3/8/2022	3/8/2022	Process of requirement engineering, requirement elicitation Methods	Slides, Lab experiment
3	8/8/2022	8/8/2022	Requirement modelling	Slides, Lab experiment
4	10/8/2022	10/8/2022	SRS IEEE Format	Slides, Lab experiment
5	12/8/2022	12/8/2022	DFD and Behavioral Models	Slides, Lab experiment

Module 3: Software Estimation Metrics				
Lecture No.	**Date**		**Topic**	**Content Delivery Method**
	Planned	**Actual**		
1	17/8/2022	17/8/2022	Software project estimation, size-oriented metrics, Function point	Slides, Lab experiment
2	22/8/2022	22/8/2022	Function point contd.., LOC calculations, numerical on FP and LOC	Slides, Lab experiment
3	23/8/2022	23/8/2022	COCOMO model	Slides, Lab experiment
4	24/8/2022	24/8/2022	Intermediate and Detailed Cocomo	Slides, Lab experiment
5	25/8/2022	25/8/2022	Project scheduling, principles of scheduling, creating a task network	Slides, Lab experiment
6	29/8/2022	29/8/2022	Critical path Method	Slides, Case Study
7	12/9/2022	12/9/2022	Program Evaluation and Review Technique(PERT)	Slides, Case Study
8	13/9/2022	13/9/2022	Earned Value Analysis	Slides, Case Study

Module 4: Software Design				
Lecture No.	**Date**		**Topic**	**Content Delivery Method**
	Planned	**Actual**		
1	14/9/2022	14/9/2022	Software Design principles	Slides, Lab Experiment
2	19/9/2022	19/9/2022	Software design concepts	Slides, Lab Experiment
3	20/9/2022	20/9/2022	software design concepts continued, Coupling and cohesion	Slides
4	21/9/2022	21/9/2022	architectural design types, Interface design	Slides
5	23/9/2022	23/9/2022	Effective Modular Design	Slides

Module 5: Software Testing				
Lecture No.	**Date**		**Topic**	**Content Delivery Method**
	Planned	**Actual**		
1	26/9/2022	26/9/2022	software testing, unit testing, integration testing, regression testing	Slides, Lab Experiment, case study
2	27/9/2022	27/9/2022	validation testing, system testing, intro to white box testing	Slides, Lab Experiment, case study
3	28/9/2022	28/9/2022	basis path testing, control structure testing	Slides, Lab Experiment, case study
4	3/10/2022	3/10/2022	Data flow testing, black box testing- boudry value and equivalence partitioning	Slides, Lab Experiment, case study
5	4/10/2022	4/10/2022	Graph based Testing	Slides
6	10/10/2022	10/10/2022	Types of Software Maintenance, Re-Engineering,	Slides
7	11/10/2022	11/10/2022	Reverse Engineering	Slides

Module 6: Software Configuration Management, Quality Assurance and Maintenance				
Lecture No.	**Date**		**Topic**	**Content Delivery Method**
	Planned	**Actual**		
1	12/10/2022	12/10/2022	What is Risk Analysis & Management? Its steps, Types of risks	Slides
2	13/10/2021	13/10/2021	risk assessment table, risk projection, Risk Mitigation, Monitoring and Management Plan (RMMM).	Slides
3	20/10/2021	20/10/2021	RIS(Risk Information Sheet), Quality Concepts and Software Quality assurance Metrics	Slides
4	21/10/2021	21/10/2021	Formal Technical Reviews, Software Reliability	Slides
5	22/10/2021	22/10/2021	The Software Configuration Management (SCM process)	Online lecture
6	22/10/2021	22/10/2021	Version Control and Change Control	Online lecture

FR. CONCEICAO RODRIGUES COLLEGE OF ENGG.
Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.

I UNIT TEST

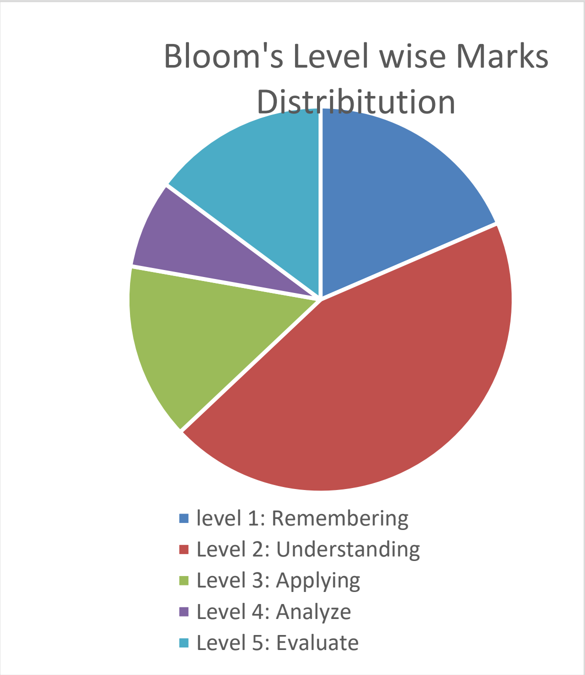
SEMESTER / BRANCH: V/ TE COMPUTER B
SUBJECT: Software Engineering (SE) MAX. MARKS: 20
DATE:5/9/2022 TIMING: 11:30am-12:30 pm

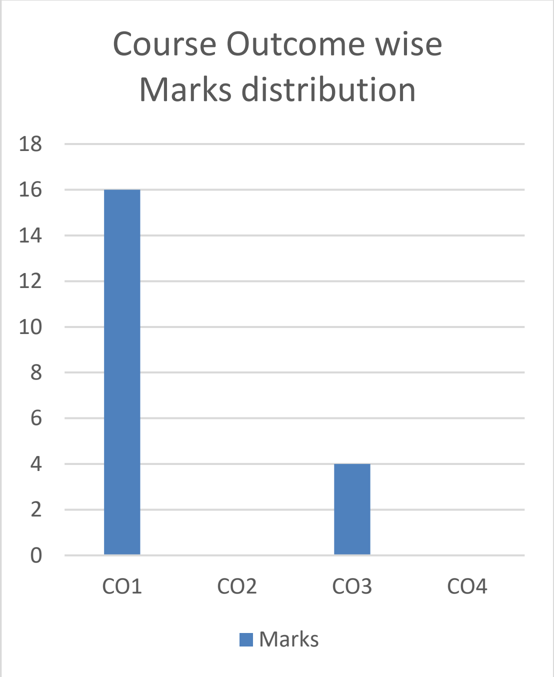
Student should be able to

CSC502.1	Identify requirements and assess the process models.
CSC502.2	Design the software Project.

Q.NO	Questions	Marks	CO	Bl	PI
1)	How is CMM different than process model? List down the levels of CMM and significance of each level through diagram **OR**	5	CO1	1	2.2.3
	Differentiate between agile and non agile process model	5	CO1	2	2.2.4
2)	The department of public works for a large city has decided to develop a Web-based pothole tracking and repair system (PHTRS). A description follows: Citizens can log onto a website and report the location and severity of potholes. As potholes are reported they are logged within a “public works department repair system” and are assigned an identifying number, stored by street address, size (on a scale of 1 to 10), location (middle, curb, etc.),				

	district (determined from street address), and repair priority (determined from the size of the pothole). Work order data are associated with each pothole and include pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used, and cost of repair (computed from hours applied, number of people, material and equipment used). Finally, a damage file is created to hold information about reported damage due to the pothole and includes citizen's name, address, phone number, type of damage, and dollar amount of damage. PHTRS is an online system; all queries are to be made interactively.				
2.A	List down the functional requirements of the system	2M	CO1	4	3.1.6
2.B	Draw a UML use case diagram PHTRS system. You'll have to make a number of assumptions about the manner in which a user interacts with this system.	2M	CO3	3	3.2.2
2.C	Develop a swim lane diagram for one or more aspects of PHTRS.	2M	CO3	3	3.2.2
2.D	Which process model will be suitable to develop this system? Justify using suitable example.	1M	CO1	5	2.3.1
3)	Describe the process of requirement Engineering in brief? (Diagram is compulsory)	7M	CO1	2	1.3.1
4)	In which situations one can apply Kanban process framework?	1M	CO1	5	3.3.1



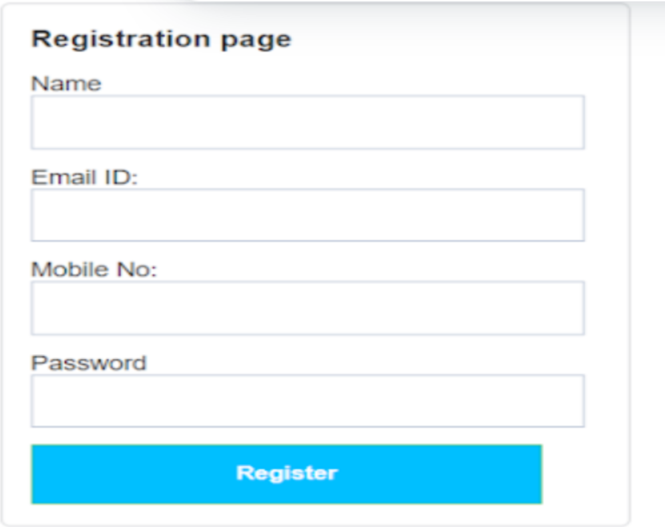


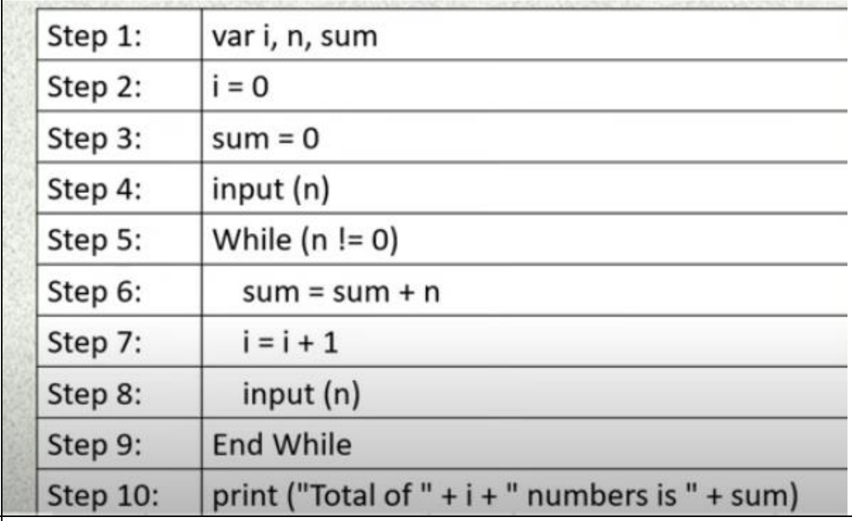
FR. CONCEICAO RODRIGUES COLLEGE OF ENGG.
Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.
II UNIT TEST

SEMESTER / BRANCH: V/ TE COMPUTER B
SUBJECT: Software Engineering (SE) MAX. MARKS: 20
DATE:17/10/2022 TIMING: 11:30am-12:30 pm

Student should be able to

CSC502.2	CO2	Plan, Schedule and track the progress of the project
CSC502.3	CO3	Design the software projects.
CSC502.4	CO4	Do testing of software project.
CSC502.5	CO5	Identify risks, manage the change to assure quality in software projects.

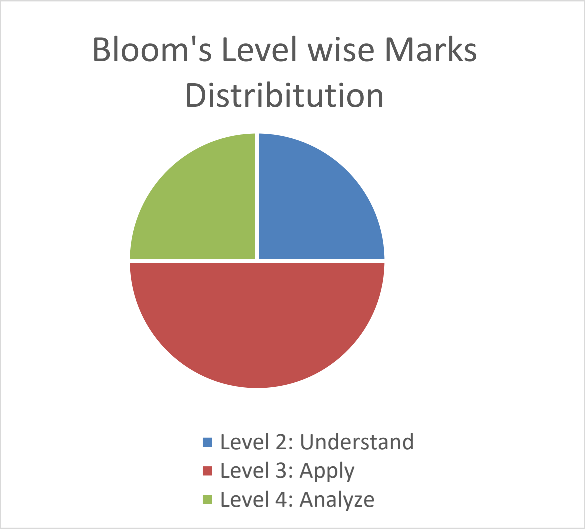
Q. N.	**Questions**	**Marks**	**CO**	**Bl**	**PI**
	Functional independence of a software design is assessed using which two criteria? highlight the differences between both.	5M	CO3	2	1.3.1
	 Registration page Name Email ID: Mobile No: Password Register Using tabular form, mention all test cases needed to perform unit testing on the **mobile number and email field** of the registration form. OR Find all DU paths for the program given below and design test cases for **All Def Coverage**	5M	CO4	3	4.1.2

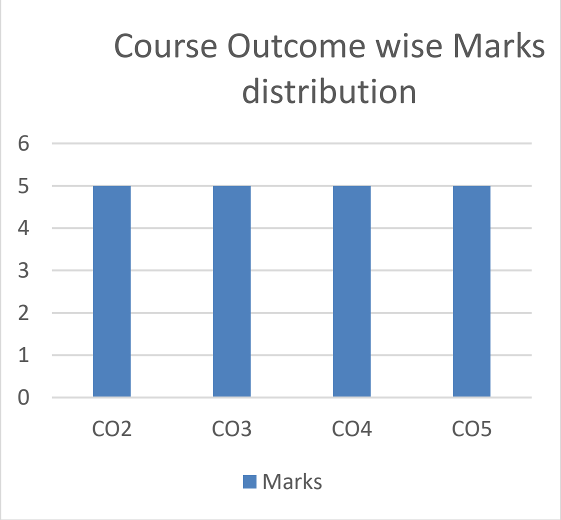


```
Step 1:  var i, n, sum
Step 2:  i = 0
Step 3:  sum = 0
Step 4:  input (n)
Step 5:  While (n != 0)
Step 6:      sum = sum + n
Step 7:      i = i + 1
Step 8:      input (n)
Step 9:  End While
Step 10: print ("Total of " + i + " numbers is " + sum)
```

3)	Prepare Risk Information sheet for High staff turnover risk. **OR** Illustrate various activities carried during software Re-engineering in brief and mention any real world software known to you which has been re- engineered.	5M	CO5	4	2.4.3
4)	A project of 200 KLOC size is to be developed. Software development team has average experience on similar type of projects. The project schedule is not very tight. Calculate the Effort, development time, average staff size, and productivity of the project using basic Cocomo.	5M	CO2	3	2.4.1

Mode	**ai**	**bi**	**ci**	**di**
organic	3.2	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	2.8	1.20	2.5	0.32





Fr. Conceicao Rodrigues College of Engineering			
Class: TE Computer B **Assignment 1** **Course Code: CSC502**			
CO1:	**Identify requirements and access the process model**		
CO2:	**Design the software projects**		
CO3:	**Plan, schedule and track the progress of the projects**		
Q. 1)	Discuss how user requirements are different than system requirements. justify using suitable example.	CO1	3.1.2 3.1.3
Q. 2)	Consider an online food ordering system like Swiggy is to be developed using an agile framework Scrum. Clearly mention what will be contents of i) Product backlog ii) Sprint backlog iii) Activities conducted by you during sprint planning iv) What do you mean by timeboxing in Scrum? When can a Sprint be cancelled and by whom? v) How can a scrum master contribute to the sprint planning process?	CO1	3.1.2 3.1.3 3.1.5
Q. 3)	Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems: ■ A system to control anti-lock braking in a car ■ A university accounting system that replaces an existing system	CO1	1.4.1 2.2.4
Q. 4)	Consider the software required to implement a full navigation capability (using GPS) in a mobile, handheld communication device. Identify two or three crosscutting concerns that would be present. Discuss how you would represent one of these concerns as an aspect.	CO2	3.3.1 3.3.2
Q. 5)	Draw an architectural diagram of a web Application and discuss in brief the purpose of each component in it	CO2	1.4.1
Q. 6)	Present two examples of applications for each of the architectural styles along with justification.	CO2	3.2.1 1.4.1
Q. 7)	Evaluate different tools available in market for software project scheduling and tracking and List down the pros and cons of each (minimum 4 tools)	CO3	2.2.3 2.2.4

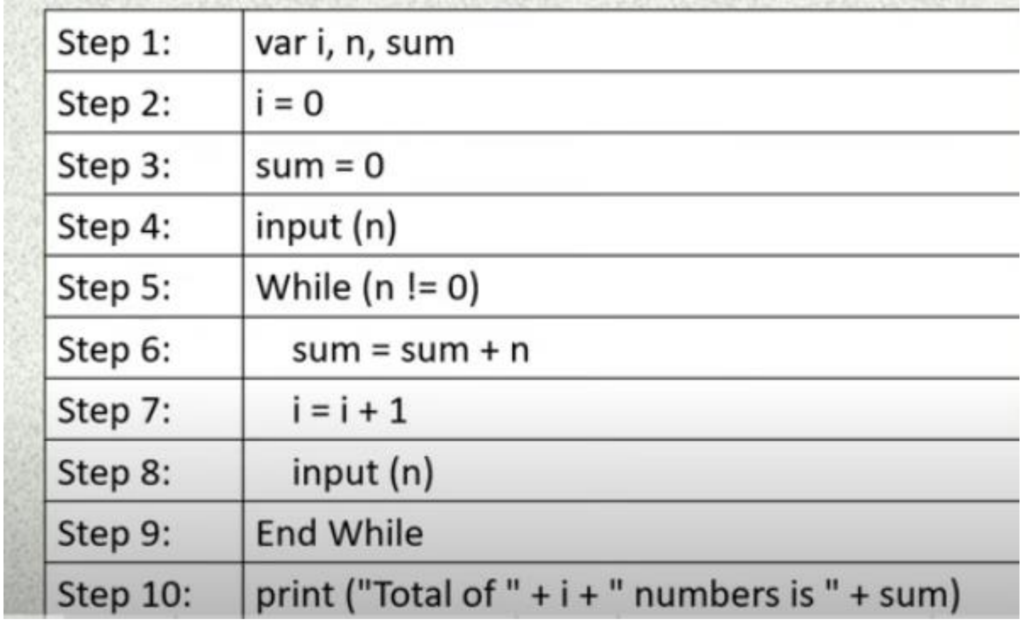
<u>Software Engineering Assignment 2</u>

YEAR: 22-23 **Class: TE COMPS B**

Q.1)	Identify any project risk which may become reality during software development. Prepare a risk information sheet for the same.	CO5	1.4.1 4.1.1

CO4: Do testing of software project.

CO5: Identify risks, manage the change to assure quality in software projects.

<table>
<tr><td>Q.2)</td><td>A safety-critical software system for treating cancer patients has two principal components: ■ A radiation therapy machine that delivers controlled doses of radiation to tumor sites. This machine is controlled by an embedded software system
■ A treatment database that includes details of the treatment given to each patient. Treatment requirements are entered in this database and are automatically downloaded to the radiation therapy machine.
Identify three hazards that may arise in this system. For each hazard, suggest a defensive requirement that will reduce the probability that these hazards will result in an accident. Explain why your suggested defense is likely to reduce the risk associated with the hazard</td><td>CO5</td><td>4.3.2</td></tr>
<tr><td>Q.3)</td><td>Select a software component that you have designed and implemented recently. Design a set of test cases to perform white box and black box testing on the same.</td><td>CO4</td><td>4.1.2</td></tr>
<tr><td>Q.4)</td><td>Research an existing SCM tool and describe how it implements the mechanics of ve control.</td><td>CO5</td><td>5.1.2</td></tr>
<tr><td>Q.5)</td><td>Design Test cases to perform Data Flow Testing for below program

<table>
<tr><td>Step 1:</td><td>var i, n, sum</td></tr>
<tr><td>Step 2:</td><td>i = 0</td></tr>
<tr><td>Step 3:</td><td>sum = 0</td></tr>
<tr><td>Step 4:</td><td>input (n)</td></tr>
<tr><td>Step 5:</td><td>While (n != 0)</td></tr>
<tr><td>Step 6:</td><td>   sum = sum + n</td></tr>
<tr><td>Step 7:</td><td>   i = i + 1</td></tr>
<tr><td>Step 8:</td><td>   input (n)</td></tr>
<tr><td>Step 9:</td><td>End While</td></tr>
<tr><td>Step 10:</td><td>print ("Total of " + i + " numbers is " + sum)</td></tr>
</table></td><td>CO4</td><td>4.1.2</td></tr>
<tr><td>Q. 6)</td><td>Describe the need of software Re engineering and briefly discuss about the activitie carried during it.</td><td>CO5</td><td>1.4.1</td></tr>
</table>

Submitted By	**Approved By**
Prof. Ashwini Pansare	ii) Dr. Sujata Deshmukh Sign:
Sign:	ii) Dr. B. S. Daga Sign:
	iii) Prof. Merly Thomas Sign:
	iv) Prof. Roshni Padate Sign:
	v) Prof. Kalpana Deorukhkar Sign:

Date of Submission:	**Date of Approval:**
Remarks by DQAC (if any)	