INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad -500 043

INFORMATION TECHNOLOGY

COURSE DESCRIPTOR

Course Title	**SOFTWARE ENGINEERING LABORATORY**				
Course Code	ACS107				
Programme	B.Tech				
Semester	IV	IT			
	V	CSE			
Course Type	Core				
Regulation	IARE - R16				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	3	1	4	3	2
Chief Coordinator	Mr. A. Praveen, Assistant Professor, IT				
Course Faculty	Mr. A. Praveen, Assistant Professor, IT Ms. B.Dhana Laxmi, Assistant Professor, IT				

I. COURSE OVERVIEW:

The aim of this course is to select and analyze the suitable software development process model for the given scenario. Classify the requirements and prepare software requirement documents for analyzing the projects. It also helps in understanding the different design techniques and their implementation process in detail. Distinguishes various testing methodologies for verification and validation process of different design models using different case studies.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites
UG	-	-	Basic Knowledge of Computer Software's

III. MARKS DISTRIBUTION:

Subject	SEE Examination	CIA Examination	Total Marks
Software Engineering Laboratory	70 Marks	30 Marks	100

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

✘	Chalk & Talk	✘	Quiz	✘	Assignments	✘	MOOCs
✔	LCD / PPT	✘	Seminars	✘	Mini Project	✔	Videos
✔	Open Ended Experiments						

V. EVALUATION METHODOLOGY:

Each laboratory will be evaluated for a total of 100 marks consisting of 30 marks for internal assessment and 70 marks for semester end lab examination. Out of 30 marks of internal assessment, continuous lab assessment will be done for 20 marks for the day to day performance and 10 marks for the final internal lab assessment.

Semester End Examination (SEE): The semester end lab examination for 70 marks shall be conducted by two examiners, one of them being Internal Examiner and the other being External Examiner, both nominated by the Principal from the panel of experts recommended by Chairman, BOS.

The emphasis on the experiments is broadly based on the following criteria:

20 %	To test the preparedness for the experiment.
20 %	To test the performance in the laboratory.
20 %	To test the calculations and graphs related to the concern experiment.
20 %	To test the results and the error analysis of the experiment.
20 %	To test the subject knowledge through viva – voce.

Continuous Internal Assessment (CIA):
CIA is conducted for a total of 30 marks (Table 1), with 20 marks for continuous lab assessment during day to day performance, 10 marks for final internal lab assessment.

Table 1: Assessment pattern for CIA

Component	**Laboratory**		**Total Marks**
Type of Assessment	Day to day performance	Final internal lab assessment	
CIA Marks	20	10	30

Continuous Internal Examination (CIE):
One CIE exams shall be conducted at the end of the 16th week of the semester. The CIE exam is conducted for 10 marks of 3 hours duration.

Preparation	**Performance**	**Calculations and Graph**	**Results and Error Analysis**	**Viva**	**Total**
4	4	-	-	2	10

VI. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		Strength	Proficiency assessed by
PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3	Videos
PO 2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	3	Case Studies
PO 3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	2	Videos
PO 4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	2	Case Studies

3 = High; 2 = Medium; 1 = Low

VII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSOs)		Strength	Proficiency assessed by
PSO 1	**Professional Skills:** The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based systems of varying complexity.	2	Videos
PSO 2	**Software Engineering Practices:** The ability to apply standard practices and strategies in software service management using open-ended programming environments with agility to deliver a quality service for business success.	2	Case Studies
PSO 3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.	1	Case Studies

3 = High; 2 = Medium; 1 = Low

VIII. COURSE OBJECTIVES (COs):

The course should enable the students to:	
I	Select suitable software development process model for the given scenario.
II	Classify the requirements and prepare software requirements documents for analyzing the projects.

III	Understand the different design techniques and their implementation.
IV	Apply various testing methodologies for validating design models.

IX. COURSE LEARNING OUTCOMES (CLOs):

CLO Code	CLO's	At the end of the course, the student will have the ability to:	PO's Mapped	Strength of Mapping
ACS107.01	CLO 1	Understand the role of software	PO 1	2
ACS107.02	CLO 2	Determine the problems occurred due to various software crisis.	PO 1, PO 3	2
ACS107.03	CLO 3	Understand the need of requirements engineering process.	PO 1, PO 3	2
ACS107.04	CLO 4	Compare the process of requirements development and requirements management.	PO 1, PO 2, PO 4	2
ACS107.05	CLO 5	Determine the importance of requirements classification.	PO 1, PO 3	2
ACS107.06	CLO 6	Understand the difference between verification and validation process.	PO 1, PO 2, PO 4	2
ACS107.07	CLO 7	Determine the principle of design stating high cohesion and low coupling.	PO 1, PO 2, PO 3	1
ACS107.08	CLO 8	Understand the basic principles of architectural design process.	PO 1, PO 2, PO 3	1
ACS107.09	CLO 9	Understand the different categories of testing before deployment of software.	PO 1, PO 2	3
ACS107.10	CLO 10	Determine the procedure of regression testing.	PO 1, PO 3	2
ACS107.11	CLO 11	Understand the importance of performance testing.	PO 1, PO 3	2
ACS107.12	CLO 12	Determine the concepts of software metrics used before software deployment.	PO 1, PO 2	3

3 = High; 2 = Medium; 1 = Low

X. MAPPING COURSE LEARNING OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Learning Outcomes (CLOs)	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CLO 1	3												1	2	
CLO 2	3		3										1		
CLO 3	3		3										1	2	
CLO 4	2	2		2									1	2	
CLO 5	2		2										1	2	
CLO 6	2	2		2										2	
CLO 7	1	1	1										1		
CLO 8	1	1	1											2	

Course Learning Outcomes (CLOs)	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CLO 9	2	2												2	
CLO 10	2		2										1		
CLO 11	3		3											2	
CLO 12	3	3											1		

3 = High; 2 = Medium; 1 = Low

XI. ASSESSMENT METHODOLOGIES – DIRECT

CIE Exams	PO 1, PO 2 PO 3, PO 4	SEE Exams	PO 1, PO 2 PO 3, PO 4	Assignments	-	Seminars	PO 1, PO 2 PO 3, PO 4
Laboratory Practices	PO 1, PO 2 PO 3, PO 4	Student Viva	PO 1, PO 2 PO 3, PO4	Mini Project	-	Certification	-

XII. ASSESSMENT METHODOLOGIES - INDIRECT

✔	Early Semester Feedback	✔	End Semester OBE Feedback
✘	Assessment of Mini Projects by Experts		

XIII. SYLLABUS

WEEK-l	ROLE OF SOFTWARE
Background: Software has made the world a global village today. The impact of software spans across almost all aspect of human life. All organizations, Institutions and companies are leveraging the potentials of software in automating the critical functions and eliminating manual interventions. Software is also a predominant are for trade and export especially for the countries like India. Domains like health care, Airlines , financial Services, Insurance , retails, Education, and many more have exploited software and still there a lot of the scope for software to create impact and add values in multiple dimensions. **Problem Description:** In the context of this background, identify the areas (or application or systems) how software has been leveraged extensively in the following domains 1. Health Care 2. Airlines 3. Banking Insurance 4. Retail 5. Education	
WEEK-2	**SOFTWARE CRISIS**
Background: In the early years of computers applications, the focus of the development and innovation were on hardware. Software was largely views as an afterthought. Computer programming was an art. Programmers did not follow any disciplined or formalized approaches. This way of doing things was adequate for a while, until the sophisticated of computer applications	

outgrow. Software soon took over and more functions which were done manually. A software houses begin to develop for widespread distribution. Software development projects produced thousands of source program statement. With the increase in the size and complexity of the software, following situation resulted is collectively termed as software crisis.

1. Time Slippage
2. Cost Slippage
3. Failure at customer Site
4. Intractable Error after delivery

Problem Description: In the context of this background, for each of the scenario mentioned below, identify the most appropriate problem related to software crisis and mention the same in the table provided.

Background: In the early years of computers applications, the focus of the development and innovation were on hardware. Software was largely views as an afterthought. Computer programming was an art. Programmers did not follow any disciplined or formalized approaches. This way of doing things was adequate for a while, until the sophisticated of computer applications outgrow. Software soon took over and more functions which were done manually. A software houses begin to develop for widespread distribution. Software development projects produced thousands of source program statement. With the increase in the size and complexity of the software, following situation resulted is collectively termed as software crisis.

1. Time Slippage
2. Cost Slippage
3. Failure at customer Site
4. Intractable Error after delivery

Problem Description: In the context of this background, for each of the scenario mentioned below, identify the most appropriate problem related to software crisis and mention the same in the table provided.

Scenario A: Railways reservation software was delivered to the customer and was installed in one of the metro station at 12.00 AM (mid-night) as per the plan. The system worked quite fine till the next day 12.00 PM (noon). The system crashed at 12.00 PM and the railways authorities could not continue using software for reservation till 02.00 PM. It took two hours to fix the defect in the software in the software.

Scenario B: A polar satellite launch vehicle was scheduled for the launch on August 15th. The auto-pilot of the rocket to be delivered for integration of the rocket on may 15th. The design and development of the software for the auto-pilot more effort because of which the auto-pilot was delivered for the integration on June 15th (delayed by a month). The rocket was launched on Sep 15th (delayed by a month).

Scenario C: Software for financial systems was delivered to the customer. Customer informed the development team about a mal-function in the system. As the software was huge and complex, the development team could not identify the defect in the software.

Scenario D: Due to the defect in the software for the baggage handling system. There was also loss of $2M of revenues for the airport authorities.

Scenario	Situation (as given A to D)
A	
B	
C	
D	

WEEK-3 REQUIREMENT DEVELOPMENT

Background: Requirement engineering produces a specification of what a system should do. The intention of requirement engineering is to provide a clear definition of requirement of the systems. This phase is a very important phase because, if the customer requirements are not clearly understood, the ambiguity can get into the other phase of the development. To avoid such issues, requirement has to be elicited using the right elicitation techniques, to be analyzed effectively, specified clearly and verified thoroughly. All activities are collectively termed as requirement development activities.

Problem Description: Identify the requirement development activities associated with each of the following scenarios,

a. Joe is creating an online survey questionnaire for requesting user feedback on the desired features of the application to be developed.
b. Mark is preparing a formal document which includes all of the desired features identified by the survey.
c. Jack identified an incomplete requirement statement
d. Jones is identifying all security related requirement and separating them from the performance related requirements
e. Merlin a team member is sent to client to observe the business case and collect typical user requirements
f. Leo is team member is working on requirement and ensuring that requirement collected should not be vague and unclear.
g. Lee is conducting a facilitated meeting with the stakeholder to capture the requirements.
h. Amit a team member is distributing questionnaires to stack holder for gathering user requirements.

Scenario	**Requirement Development Activities**
A	
B	
C	
D	
E	
F	
G	
H	

WEEK-4 REQUIREMENT CLASSIFICATION AND VERIFICATION

A. **Background:** Functional requirements (FRs) specify the software functionality that the developer must build into the product to enable users accomplish their tasks, thereby satisfying the business requirements. Nonfunctional requirement as the name suggest, are those requirements which are not directly concerned with the specific functions delivered by the system. Many non-functional requirements (NFRs) related to the system as a whole rather than to individual functional requirements. While failure to meet an individual functional may degrade the system, failure to meet a non-functional system requirement may make whole system unusable. NFR's are of di reliability requirements etc.

Problem Description: Classify the following requirement by selecting the appropriate option.

1. ATM machine shall validate PIN of the user during login along with bio-metric verification.
2. "Peak transaction-20,000calls in Volume(s)abusyhour, average duration 20 Secs, grade of services 99.98%.
3. "Brahe System sounds the alarmShallfor10seconds at frequency of 100H when the brake is applied".
4. "Mean Time Failure (MTTF) to -There should be no more than three Severity-1 outage per month".

B. **Background:** Software requirements specification formally captures the requirements of the software to be developed. Hence it is important that requirements are free from defects like incorrect or conflicting requirements.

Problem Description: Identify the requirements in the given SRS(Premium University Placement Portal) for following issues,

1. Incorrect requirements
2. Ambiguous requirements
3. Missing requirements
4. Conflicting requirements
5. Incomplete requirements

WEEK-5 SOFTWARE DESIGN PRINCIPLES

Background: A good object oriented design not only meets the specified requirements but also addresses implicit requirements. There are five design principles which address most of the implicit requirements:

Software Design Principles:

1 Abstraction: Focus on solving a problem by considering the relevant details and ignoring the irrelevant

2 Encapsulation: Wrapping the internal details, thereby making these details inaccessible. Encapsulation separates interface and implementation, specifying only the public interface to the clients, hiding the details of implementation.

3 Decomposition and Modularization: Dividing the problem into smaller, independent, interactive subtasks for placing different functionalities in different components

4 Coupling & Cohesion: Coupling is the degree to which modules are dependent on each other. Cohesion is the degree to which a module has a single, well defined task or responsibility. A good design is one with loose coupling and strong cohesion.

5 Sufficiency, Completeness and Primitiveness: Design should ensure the completeness and sufficiency with respect to the given specifications in a very simple way as possible.

Problem Description: Which of the following design principle(s) have been violated in the following scenarios?

1. Abstraction
2. Decomposition and Modularization
3. Coupling & Cohesion
4. Encapsulation
5. Sufficiency, Completeness and Primitiveness
6. All

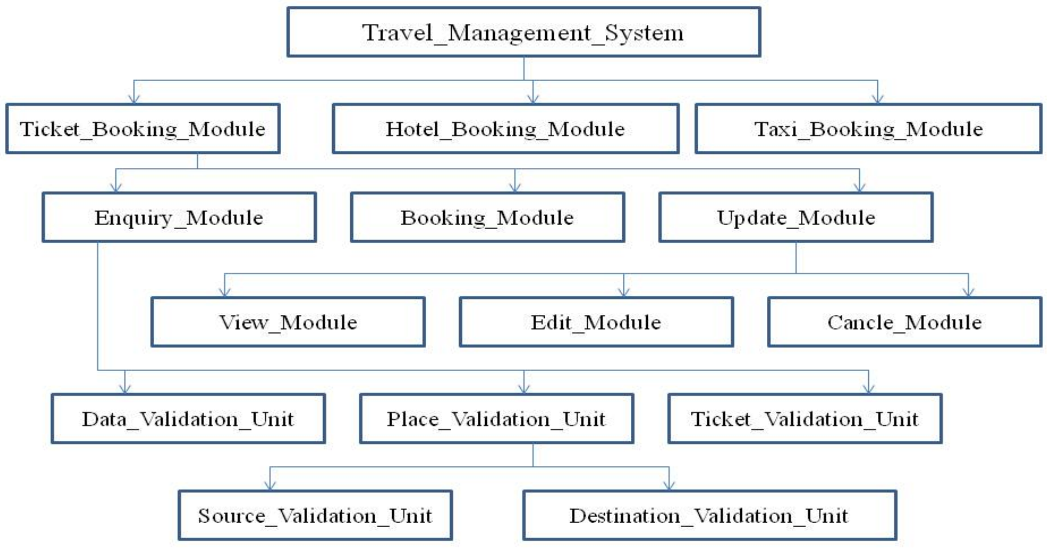
S No	Description	Principle Being Violated
1	Important information of a module is directly accessible by other modules.	
2	Too many global variables in the program after implementing design	
3	Code breaks in unexpected places	
4	Unfulfilled requirements in the code after the design has been implemented	
5	Cyclic dependencies among classes	
6	Huge class doing too many unrelated operations	
7	Several unrelated functionalities/tasks are carried out by a single module	
8	All data of all classes in public	
9	Design resulting in spaghetti code	
10	An algorithm documented as part of design is not understandable by the programmers	

WEEK-6 INTEGRATION TESTING

Background: Integration testing is carried out after the completion of unit testing and before the software is delivered for system testing. In top down integration testing, dummy stubs are required for bottom level modules. Similarly in bottom up testing, dummy drivers are required for top level modules.

Problem Description: Consider the scenario of development of software for Travel; Management System (TMS) is in progress. The TMS software has 3 major modules namely Ticket_Booking_Module, Hotel_Booking_Module and Taxi_Booking_Module. The Ticket_Booking_Module has 3 sub modules namely Enquiry_Module, Booking_Module and

Update_Module. The enquiry module uses Date_Validation_Unit, Ticket_Validation_Unit and Place_Validation_Unit.



In the context of the given scenario, identify the usage of stub or driver for the following situations.

1. Except the Ticket_validation_Unit, the coding and unit testing of all other modules, sub modules and units of TMS are completed. The top-down integration is in progress for the TMS software. To carry out the integration testing, which among the following is necessary?
   - A Stub for Ticket_Validation_Unit
   - A Driver For Ticket_Validation_Unit
   - A Stub for Enquiry_Module
   - A Driver for Enquiry_Module
   - A Stub For Ticket_Booking_Module
   - A Driver For Ticket_Booking_Module

2. The coding and unit testing of all the module, sub modules and units of TMS are completed except the Update_Module (coding and testing for Edit_Module, Cancel_Module and View_Module are also completed). The bottom-up integration is to be started for the TMS software. Mention any stub or driver needed to carry out the integration testing?
3. Except the Taxi_Booking_Module, the coding and unit testing of all other modules, sub modules and units of TMS are completed. The top-down integration is to be started for the TMS software. Mention any stub or driver needed to carry out the integration testing.

WEEK-7	**PERFORMANCE TESTING**

Background: Performance testing tests the non-functional requirements of the system. The different types of performance testing are load testing, stress testing, endurance testing and spike testing.
Problem Description: Identify the type of performance testing for the following:

1. A space craft is expected to function for nearly 8 years in space. The orbit control system of the spacecraft is a real-time embedded system. Before the launch, the embedded software is to be tested to ensure that it is capable of working for 8 years in the space. Identify the suitable performance testing category to be carried out to ensure that the space craft will be functioning for 8 years in the space as required.
2. Global Education Centre (GEC) at Infosys Mysore provides the training for fresh entrants. GEC uses an automated tool for conducting objective type test for the trainees. At a time, a maximum of 2000 trainees are expected to take the test. Before the tool is deployed, testing of the tool was carried out to ensure that it is capable of supporting 2000 simultaneous users. Indicate the performance testing category?
3. A university uses its web based portal for publishing the results of the students. When the results of an examination were announced on the website recently on a pre-planned date, the web site

crashed. Which type of performance testing should have been done during web-site development to avoid this unpleasant situation?

4. During unexpected terrorist attack, one of the popular websites crashed as many people logged into the web-site in a short span of time to know the consequences of terrorist attack and for immediate guidelines from the security personnel. After analyzing the situation, the maintenance team of that website came to know that it was the consequences of unexpected load on the system which had never happened previously. Which type of performance testing should have been done during web-site development to avoid this unpleasant situation?

Scenarios	Performance Testing Type
Scenario 1	
Scenario 2	
Scenario 3	
Scenario 4	

WEEK-8 REGRESSION TESTING

Background: Enhancements are introduction of new features to the software and might be released in different versions. Whenever a version is released, regression testing should be done on the system to ensure that the existing features have not been disturbed.

Problem Description: Consider the scenario of development of software for Travel Management System (TMS) discussed in previous assignment. TMS has been developed by Infosys and released to its customer Advance Travel Solutions Ltd. (ATSL). Integration testing, system testing and acceptance testing were carried out before releasing the final build to the customer. However, as per the customer feedback during the first month of usage of the software, some minor changes are required in the Enquiry Module of the TMS. The customer has approached Infosys with the minor changes for upgrading the software. The development team of Infosys has incorporated. Those changes, and delivered the software to testing team to test the upgraded software. Which among the following statement is true?

1 Since minor changes are there, integration of the Enquiry Module and quick system testing on Enquiry module should be done.

2 The incorporation of minor changes would have introduced new bugs into other modules, so regression testing should be carried out.

3 Since the acceptance testing is already carried out, it is enough if the team performs sanity testing on the Enquire module.

4 No need of testing any module.

WEEK-9 SOFTWARE METRICS

Background: There are some metrics which are fundamental and the rest can be derived from these. Examples of basic (fundamental) measures are size, effort, defect, and schedule. If the fundamental measures are known, then we can derive others. For example if size and effort are known, we can get Productivity (=size/effort). If the total numbers of defects are known we can get the Quality (=defect/size) and so on.

Problem Description: Online loan system has two modules for the two basic services, namely Car loan service and House loan service.
The two modules have been named as Car_Loan_Module and House_Loan_Module.
Car_Loan_Module has 2000 lines of uncommented source code. House_Loan_Module has 3000 lines of uncommented source code. Car_Loan_Module was completely implemented by Mike.
House_Loan_Module was completely implemented by John. Mike took 100 person hours to implement Car_Loan_Module. John took 200 person hours to implement House_Loan_Module.
Mike’s module had 5 had 6 defects. With respect to the context given, which among the following is an INCORRECT statement?

Choose one:

1. John’s quality is better than Mike
2. John’s productivity is more than Mike

3. John introduced more defects than Mike 4. John’s effort is more than Mike
TEXT BOOK:
1. Roger S. Pressman, “Software Engineering: A Practitioner’s Approach”, Tata McGraw Hill International Edition, 7th Edition, 2009.
REFERENCE BOOK:
1. Ian Somerville, “Software Engineering”, Pearson Education, 8th Edition, 2008.

XIV. COURSE PLAN:

The course plan is meant as a guideline. Probably there may be changes.

Week No.	Topics to be covered	Course Learning Outcomes (CLOs)	Reference
1	Role of software	CLO 1, CLO 2, CLO 3, CLO 6	T1:1.2
2	Software crisis	CLO 1, CLO 2, CLO 3, CLO 6	R1:2.4
3	Requirement engineering process	CLO 5, CLO 6, CLO 11, CLO 12	R1:2.5
4	Requirements classification	CLO 4, CLO 6	T1:2.6
5	Requirements verification	CLO 12, CLO 6	R1:22.7
6	Software design principles	CLO 7, CLO 10, CLO 11, CLO 12	R1:5.3
7	High cohesion and low coupling	CLO 7, CLO 10, CLO 11, CLO 12	T1:6.3
8	Requirements development process	CLO 8 CLO 10, CLO 11, CLO 12	R1:6.8
9	Testing concepts: Integration testing	CLO 8, CLO 9, CLO 11	R1:13.1
10	Testing concepts: Performance testing	CLO 8, CLO 9, CLO 11	T1:13.2
11	Testing concepts: Regression testing	CLO 8, CLO 9, CLO 11	R1:13.7
12	Software Metrics	CLO 10, CLO 11, CLO 12	T1:10.2

XV. GAPS IN THE SYLLABUS - TO MEET INDUSTRY / PROFESSION REQUIREMENTS:

S No	Description	Proposed Actions	Relevance With POs	Relevance With PSOs
1	How to collect useful requirements to build right product	Seminars/Guest Lectures / NPTEL	PO 1, PO 2, PO 3	PSO 1, PSO 2
2	Real time Risk management System	Seminars/Guest Lectures / NPTEL	PO 2, PO 3	PSO 1
3	Generation of test cases for usage of ATM machine and Banking Applications	Assignments/ Laboratory Practices	PO 1, PO 3, PO 4	PSO 2

Prepared by:
Mr. A.Praveen, Assistant Professor, IT
Ms. B.Dhana Laxmi, Assistant Professor, IT

HOD, INFORMATION TECHNOLOGY