HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

HONORS AND MINORS SPECIALIZATIONS

2018 REGULATIONS

HONORS APPLICABLE FOR B.Tech., B.Des, SLAAS PROGRAMMES

MINORS APPLICABLE FOR ALL PROGRAMMES

UNDER

HITS – COURSERA SKILL TRANSFORMATION PROGRAMME – HCST

1. **HONOURS COURSES THROUGH COURSERA**
2. **HONOURS COURSES OFFERED BY DEPT**
3. **MINORS OFFERED THROUGH COURSERA**
4. **MINORS OFFERED BY DEPARTMENT**
5. **NON DEPARTMENT ELECTIVES THROUGH COURSERA**
6. **SPECIALIZATIONS, VALUE ADDED COURSES THROUGH COURSERA**

7.0 B. Tech, / B.Des., / SLAAS (Honours) Programme

A new academic programme B. Tech, / B.Des., / SLAAS. (Hons.) is introduced in order to facilitate the students to choose additionally the specialized courses of their choice and build

their competence in a specialized area. The features of the new programme, include:

a. B.Tech. students in regular stream can opt for B. Tech, / B.Des., / SLAAS (Hons) provided they have a CGPA of 8.0 up to the end of fourth semester without any history of arrears.
b. The students opting for this program have to take four additional courses of their specialization of a minimum of 3 credits each from 5th to 8th semesters with not more than 2 additional courses in a semester.
c. The list of such additional courses offered by the various Departments of the respective school will be announced in the beginning of the academic year to facilitate the registration process.
d. The student can also opt for post graduate level courses
e. The faculty advisor will suggest the additional courses to be taken by the students based on their choice and level of their academic competence.
f. Students who have obtained "E" or "U" or "RA" grade or "DE" category (vide clause 16.0 – Grading) in any course, including the additional credit courses, are not eligible for B. Tech, / B.Des., / SLAAS (Hons) degree.
g. The students have to pay the requisite fee for the additional courses.

- *A student can opt for only one Honors Specialization for the entire duration of the programme.*
- *The successful student will be awarded the Degree Certificate with Hons specialization namely, "B.Tech in Computer Science and Engineering with Hons specialization in Network Security"*

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

8.0 Minor specialization:

Students, who are desirous of pursuing their special interest areas other than the chosen discipline of Engineering / Technology/ Arts/ Fashion/ Humanities/ Management/ Basic Sciences, may opt for additional courses in minor specialization groups offered by a department other than their parent department. Such students shall select the stream of courses offered with pre – requisites by the respective departments and earn a Minor Specialization.

a. The number of credits to be earned for Minor specialization is 12 credits.
b. The students are permitted to register for their minor specialization courses from the V semester onwards subject to a maximum of two additional courses per semester.
c. The list of such additional courses offered by the various departments and the schedule will be announced in the beginning of the academic year to facilitate the registration process.
d. The students have to pay the requisite fee for the additional courses.

- *A student can opt for multiple Minor specializations.*
- *The student will be awarded Minor Specialization certificates along with the Degree Certificate upon successful completion.*

The list of Hons and Minors, NE, Specializations offered through Coursera and the Departments are listed in this document.

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

HONOURS

GROUP 1

SCHOOL OF COMPUTING SCIENCE & SCHOOL OF ELECTRICAL SCIENCE

COURSERA:

1. PRODUCT AND SOFTWARE MANAGEMENT
2. COMPUTER ARCHITECTURE AND NETWORKS
3. CYBERSECURITY
4. DATA ENGINEERING ON CLOUD
5. BUSINESS ANALYTICS
6. TECHNOLOGY AND MIS MANAGEMENT
7. DATA STRUCTURES AND ALGORITHM
8. BIG STRUCTURES AND DATABASE SYSTEMS
9. DEEP LEARNING
10. INTERNET OF THINGS
11. MOBILE COMPUTING
12. TENSORFLOW
13. WEB DEVELOPMENT
14. WIRELESS SENSOR NETWORK GAME PROGRAMMING

DEPARTMENT OFFERED HONOURS:

1. SMART GRID
2. IOT EMBEDDED SYSTEMS
3. EMBEDDED SYSTEMS
4. ARTIFICIAL INTELLIGENCE
5. VIRTUALIZATION
6. ELECTRIC VEHICLES DESIGN

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

GROUP 2

SCHOOL OF MECHANICAL SCIENCE & SCHOOL OF AERONAUTICAL SCIENCE

COURSERA

1. DIGITAL MANUFACTURING
2. ENERGY MANAGEMENT IN INDUSTRY
3. MATERIAL SCIENCE
4. MECHANICS OF MATERIALS
5. THERMODYNAMICS
6. VIBRATION AND ACOUSTICS
7. FOOD MICROBIOLOGY
8. GENETICS
9. PHARMACEUTICAL BIOTECHNOLOGY
10. PROJECT MANAGEMENT
11. RELIABILITY AND QUALITY CONTROL
12. SELF DRIVING CARS
13. SUPPLY CHAIN AND PROCUREMENT

DEPARTMENT OFFERED HONOURS

1. DESIGN
2. ENVIRONMENTAL HEALTH ENGINEERING
3. PROCESS ENGINEERING
4. GENETIC ENGINEERING18.ROBOTICS
5. AIRCRAFT STRUCTURES
6. AERODYNAMICS
7. PROPULSION
8. AUTOTRONICS
9. MOTORSPORTS

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

GROUP 3

SCHOOL OF BUILDING SCIENCE & SCHOOL OF DESIGN

COURSERA

1. ENGINEERING ECONOMICS AND MANAGEMENT
2. ENVIRONMENTAL ENGINEERING
3. SMART CITY AND INFRASTRUCTURAL ENGINEERING
4. SUSTAINABLE AND RENEWABLE ENERGY
5. WATER RESOURCE ENGINEERING

DEPARTMENT OFFERED HONOURS

1. ENVIRONMENTAL SYSTEMS
2. CONSTRUCTION MANAGEMENT
3. DISASTER MANAGEMENT

Honors in Embedded systems

COURSE TITLE	IT AUTOMATION WITH PYTHON - I			CREDITS	3
COURSE CODE	ECH4376	COURSE CATEGORY	Honors	L-T-P-S	2-0-2-0
Version	1.0	Approval Details		LEARNING LEVEL	BTL-3

ASSESSMENT SCHEME

First Periodical Assessment	Second Periodical Assessment	Practical Assessment	ESE
15%	15%	20%	50%

Course Description	This course covers the basic foundations to write simple programs in Python using the most common structures.
Course Objective	1. To familiarize the students with basic Python structures: strings, lists, and dictionaries 2. To write simple python scripts using the basic syntax 3. To develop simple problem statement and write program to solve them 4. To manage files and formulate regular expressions in Python 5. To write programs for simple real time problems
Course Outcome	Upon completion of this course, the students will be able to 1. Interpret the basic Python structures: strings, lists, and dictionaries 2. Write short Python scripts to perform automated actions and create Python objects 3. Solve simple problem statement using Python 4. Implement regular expressions (regex), and process text files 5. Develop a simple project by incorporating files, regex, and data manipulation

Prerequisites:

CO, PO AND PSO MAPPING

CO	PO -1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO - 10	PO-11	PO-12	PSO-1	PSO-2
CO-1	-	2	2	1	3	3	-	-	2	-	1	3	1	2
CO-2	-	2	1	2	3	3	-	2	2	2	2	3	1	3
CO-3	-	3	2	2	3	3	-	-	2	-	1	3	1	3
CO-4	-	1	2	2	3	3	-	-	2	-	1	3	1	2
CO-5	-	3	3	2	3	3	1	2	3	2	1	3	1	3

1: Weakly related, 2: Moderately related and 3: Strongly related

MODULE 1: Basics of Python Programming	(9)
Basic Python Syntax - Basic Python Syntax introduction - Data Types – Variables Expressions, Numbers, and Type Conversions - Defining Functions - Returning Values - The Principles of Code Reuse - Code Style - Comparing Things - Branching with if Statements - else Statements - elif Statements **Loops** - Introduction to Loops - while loop - Examples - Initializing Variables - Infinite Loops and How to Break Them - for loop - Examples - Nested for Loops – Recursion **Practices:** Experiments on Datatypes, Loops using Python	**CO-1** **BTL-3**
MODULE 2: Strings, List and Dictionaries	**(9)**
Strings, Lists and Dictionaries - Basic Structures Introduction - String - Parts of a String - Creating New Strings - More String Methods - Formatting Strings - List - Modifying the Contents of a List - Lists and Tuples - Iterating over Lists and Tuples - List Comprehensions - Dictionary - Iterating over the Contents of a Dictionary - Dictionaries vs. Lists **Practices:** Experiments on String handling, List and dictionaries using Python	**CO-2** **BTL-3**
MODULE 3: Object Oriented Programming	**(9)**
Object-oriented programming - Classes and Objects in Python - Defining New Classes - Instance Methods - Constructors and Other Special Methods - Documenting Functions, Classes, and Methods - About Jupyter Notebooks – Inheritance – Composition - Python Modules **Practices:** Experiments on Functions, classes, Jupyter Notebooks using Python	**CO-3** **BTL-3**
MODULE 4: Files and Regular Expressions	**(9)**
Managing Files with Python - Programming with Files - Reading Files - Iterating through Files - Writing Files - Working with Files - More File Information – Directories - CSV file - Reading CSV Files - Generating CSV - Reading and Writing CSV Files with Dictionaries **Regular Expressions** - Regular Expressions - Basic Matching with grep - Simple Matching in Python - Wildcards and Character Classes - Repetition Qualifiers - Escaping Characters - Regular Expressions in Action - Capturing Groups - More on Repetition Qualifiers - Extracting a PID Using regexes in Python - Splitting and Replacing **Practices:** Experiments on File management and regular Expressions using Python	**CO-4** **BTL-3**
MODULE 5: Data management and Testing	**(9)**
Managing Data and Processes - Managing Data and Processes - Reading Data interactively - Standard Streams - Environment Variables - Command-Line Arguments and Exit Status - Running System Commands in Python - Obtaining the Output of a System Command - Advanced Subprocess Management - Filtering Log Files with Regular Expressions - Making Sense out of the Data **Testing in Python** - Testing in Python - - Manual Testing and Automated Testing - Unit Tests - Writing Unit Tests in Python - Edge Cases - Additional Test Cases - Black Box vs. White Box - Other Test Types - Test-Driven Development - The Try-Except Construct - Raising Errors - Testing for Expected Errors	**CO-5** **BTL-3**

Bash Scripting - Bash Scripting - Basic Linux Commands - Redirecting Streams - Pipes and Pipelines - Signalling Processes - Creating Bash Scripts - Using Variables and Globs - Conditional Execution in Bash - While Loops in Bash Scripts- For Loops in Bash Scripts - Advanced Command Interaction - Choosing Between Bash and Python **Practices:** Experiments on Data management, Processes using Python and simple programs on Bash scripting	

TEXT BOOKS	
1.	Eric Matthes, (2019), *Python Crash Course, A Hands-On, Project-Based Introduction To Programming*, 2nd edition, pp. 1-544.
2.	Mark Lutz, (2013), *Learning Python 5ed: Powerful Object-Oriented Programming*, 5th edition, pp. 1-1600.
REFERENCE BOOKS	
1.	Paul Barry. (2016). *Head First Python 2e: A Brain-Friendly Guide*, 2nd Edition, pp.1-624.
2.	Allen Downey, Jeff Elkner and Chris Meyers. (2002), *How to Think Like a Computer Scientist Learning with Python*, Green tea press, 1st edition, pp. 1-280
E BOOKS	
1.	https://greenteapress.com/thinkpython/thinkCSpy/thinkCSpy.pdf
MOOC	
1.	https://www.coursera.org/learn/python-operating-system?specialization=google-it-automation

COURSE TITLE	IT AUTOMATION WITH PYTHON - II			CREDITS	3
COURSE CODE	ECH4377	COURSE CATEGORY	Honors	L-T-P-S	2-0-2-0
Version	1.0	Approval Details		LEARNING LEVEL	BTL-3

ASSESSMENT SCHEME

First Periodical Assessment	Second Periodical Assessment	Practical Assessment	ESE
15%	15%	20%	50%

Course Description	This course introduces the Git and GitHub, the various troubleshooting and debugging techniques, how to carry out the configuration management for large-scale machines and to automate and manage cloud instances. In addition to it, it deals with automating real-world tasks with python.
Course Objective	1. To familiarize the students with the version control tool, use and interact with GitHub, collaborate with others through remote repositories. 2. To implement the appropriate strategies to solve real-world IT problems, by analysing the root cause of problems in IT infrastructure 3. To apply the automation techniques to manage fleet of computers 4. To use Application Programming Interfaces (APIs) to interact with web services 5. To implement data serialization to send messages between running programs
Course Outcome	Upon completion of this course, the students will be able to 1. Summarize the version control tool, use and interact with GitHub, collaborate with others through remote repositories. 2. Analyze real-world IT problems and implement the appropriate strategies to solve those problems, demonstrate techniques to quickly find and solve the root cause of problems in IT infrastructure 3. Apply automation to manage fleet of computers 4. Implement Python external libraries for real time tasks. 5. Interpret data serialization to send messages between running programs

Prerequisites: GOOGLE IT AUTOMATION WITH PYTHON - I

CO, PO AND PSO MAPPING

CO	PO -1	PO -2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO -10	PO -11	PO-12	PSO-1	PSO-2
CO-1	-	2	3	1	3	3	-	2	2	-	-	3	1	2
CO-2	-	1	3	1	3	3	-	2	2	2	1	3	1	3
CO-3	-	2	3	2	3	3	-	2	2	-	-	3	1	3

CO-4	**-**	**2**	**3**	**2**	**3**	**3**	**-**	**2**	**2**	**-**	**-**	**3**	**1**	**2**
CO-5	**-**	**2**	**3**	**2**	**3**	**3**	**1**	**2**	**3**	**2**	**1**	**3**	**1**	**3**

1: Weakly related, 2: Moderately related and 3: Strongly related

MODULE 1: Introduction to Git and GitHub	**(9)**
Introduction to Version Control - Version Control - Keeping Historical Copies - Diffing Files - Applying Changes - Practical Application of diff and patch - Version Control and Automation **Using Git locally** - Using Git Locally - Skipping the Staging Area - Getting More Information About Our Changes - Deleting and Renaming Files - Undoing Changes Before Committing - Amending Commits – Rollbacks - Identifying a Commit - Creating New Branches - Working with Branches – Merging - Merge Conflicts **Working with Remotes** - Working with Remotes - Basic Interaction with GitHub - Working with Remotes - Fetching New Changes - Updating the Local Repository - The Pull-Merge-Push Workflow - Pushing Remote Branches **Collaboration** – Collaboration - A Simple Pull Request on GitHub - The Typical Pull Request Workflow on GitHub - Updating an Existing Pull Request - Squashing Changes - The Code Review Workflow - How to Use Code Reviews in GitHub - Managing Collaboration - Tracking Issues - Continuous Integration **Practices:** File Handling in GitHub	**CO-1** **BTL-3**
MODULE 2: Troubleshooting and Debugging Techniques	**(9)**
Troubleshooting concepts - Troubleshooting Concepts - Problem Solving Steps - Silently Crashing Application - Creating a Reproduction Case - Finding the Root Cause - Dealing with Intermittent Issues - Intermittently Failing Script - Binary search - Applying Binary Search in Troubleshooting - Finding Invalid Data **Slowness** – Slowness – use of resources by computers - Possible Causes of Slowness - Slow Web Server - Writing Efficient Code - Using the Right Data Structures - Expensive Loops - Keeping Local Results - Slow Script with Expensive Loop - Parallelizing Operations - Slowly Growing in Complexity - Dealing with Complex Slow Systems - Using Threads to Make Things Go Faster **Crashing Programs** - Crashing Programs - Systems That Crash - Understanding Crashing Applications - Internal Server Error -Accessing Invalid Memory - Unhandled Errors and Exceptions - Debugging a Segmentation Fault - Debugging a Python Crash - Crashes in Complex Systems - Communication and Documentation During Incidents **Managing Resources** - Managing Resources - Memory Leaks and How to Prevent Them - Managing Disk Space - Network Saturation - Dealing with Memory Leaks - Getting to the Important Tasks - Prioritizing Tasks - Estimating the Time Tasks Will Take - Communicating	**CO-2** **BTL-3**

Expectations - Dealing with Hard Problems - Proactive Practices - Planning Future Resource Usage - Preventing Future Problems **Practices:** Debugging, troubleshooting and resource management in GitHub	
MODULE 3: Configuration Management and the Cloud	**(9)**
Automating with Configuration Management - Automating with Configuration Management - The Driving Principles of Configuration Management **Deploying Puppet** - Deploying Puppet - Applying Rules Locally - Managing Resource Relationships - Organizing Your Puppet Modules - Puppet Nodes - Puppet's Certificate Infrastructure - Setting up Puppet Clients and Servers - Modifying and Testing Manifests - Safely Rolling out Changes and Validating Them **Automation in the Cloud** - Automation in the Cloud - Cloud Services Overview - Scaling in the Cloud - Evaluating the Cloud - Migrating to the Cloud - Spinning up VMs in the Cloud - Creating a New VM Using the GCP Web UI - Customizing VMs in GCP - Templating a Customized VM - Cloud Scale Deployments -Orchestration - Cloud Infrastructure as Code **Managing Cloud Instances at Scale** - Storing Data in the Cloud - Load Balancing - Change Management - Understanding Limitations - Getting Started with Monitoring - Getting Alerts When Things Go Wrong - Service-Level Objectives - Basic Monitoring in GCP - Recovering from Failure **Practices:** Hands on Cloud management	**CO-3** **BTL-3**
MODULE 4: Manipulating Images and Interacting with Web Services	**(9)**
Built-In Libraries vs. External Libraries - API - PIL for Working with Images Web Applications and Services - Data Serialization - Data Serialization Formats - More About JSON - The Python Requests Library - Useful Operations for Python Requests - HTTP GET and POST Methods **Practices:** Experiments on image manipulation and API development	**CO-4** **BTL-3**
MODULE 5: Automating Real-World Tasks with Python	**(9)**
Automatic Output Generation – Introduction - Introduction to Python Email Library - Adding Attachments - Sending the Email Through an SMTP Server - Introduction to Generating PDFs - Adding Tables and Graphics to PDFs **Practices:** Experiment on Python Email library	**CO-5** **BTL-3**
TEXT BOOKS	
1.	Bobby Iliev (2020). *Introduction to Git and Github* 1st Edition, pp.1-135.
REFERENCE BOOKS	
1.	Paul Barry. (2016). *Head First Python 2e: A Brain-Friendly Guide*, 2nd Edition, pp.1-624.

2.	Allen Downey, Jeff Elkner and Chris Meyers. (2002), *How to Think Like a Computer Scientist Learning with Python*, Green tea press, 1st edition, pp. 1-280
E BOOKS	
1.	https://github.com/bobbyiliev/introduction-to-git-and-github-ebook
MOOC	
1.	https://www.coursera.org/learn/configuration-management-cloud?=
2.	https://www.coursera.org/learn/automating-real-world-tasks-python?specialization=google-it-automation
3.	https://www.coursera.org/learn/introduction-git-github

COURSE TITLE	EMBEDDED SYSTEMS DESIGN VERIFICATION AND TEST			CREDITS	3
COURSE CODE	ECH4461	COURSE CATEGORY	Honors	L-T-P-S	3-0-0-0
Version	1.0	Approval Details		LEARNING LEVEL	BTL-3

ASSESSMENT SCHEME

First Periodical Assessment	Second Periodical Assessment	Seminar/ Assignments/ Project	Surprise Test / Quiz	Attendance	ESE
15%	15%	10%	5%	5%	50%

Course Description	This course will topics related to model, design, and implementation of structural representations of embedded systems. In addition to it, the systems are tested and the performance of the systems are also studied.
Course Objective	1. To familiarize the Embedded Systems Design Flow and its specifications 2. To perform Scheduling in Embedded Systems 3. To carry out verification of real time embedded systems using software tools 4. To explain the basic of digital testing and also the testing of embedded system hardware 5. To comprehend the testing for advanced faults in Real time Embedded Systems.
Course Outcome	Upon completion of this course, the students will be able to 1. Describe the Embedded Systems Design Flow and its specifications along with the hardware-software co-design. 2. Enumerate the various Scheduling in Embedded Systems. 3. Verify real time embedded systems using software tools. 4. Elucidate the basic of digital testing and the testing of embedded system hardware. 5. Interpret the testing for advanced faults in Real time Embedded Systems.

Prerequisites: Digital Design and Computer Architecture

CO, PO AND PSO MAPPING

CO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2
CO-1	2	1	1	-	-	-	-	-	-	1	-	2	2	3
CO-2	2	2	-	3	-	1	-	-	-	-	-	2	2	3
CO-3	2	3	1	3	2	2	2	1	-	-	1	2	2	3

CO-4	**2**	**2**	**3**	**-**	**2**	**-**	**2**	**-**	**-**	**-**	**-**	**2**	**2**	**3**
CO-5	**2**	**3**	**2**	**2**	**2**	**2**	**1**	**1**	**1**	**-**	**1**	**2**	**2**	**3**

1: Weakly related, 2: Moderately related and 3: Strongly related

MODULE 1: Introduction, Modeling, Hardware-Software Co-Design Principles and Details of Hardware Design	**(9)**
Introduction to Embedded Systems Design Flow, Formal specification and Modeling Strategies -I , Formal specification and Modeling Strategies – II, Hardware Software Co-Design , Architectural Design of Hardware - I , Architectural Design of Hardware – II	**CO-1 BTL-3**
MODULE 2: Introduction To Scheduling In Embedded Systems	**(9)**
System and Task level Timing Analysis, Uni-processor Real-time Scheduling, Multiprocessor Real-time Scheduling, Resource Allocation Strategies in Automotive Systems, Energy-aware and Fault-tolerant Real-time Scheduling	**CO-2 BTL-3**
MODULE 3: Introduction to Formal and System Verification for Embedded System	**(9)**
Introduction and Basic Operations on Temporal Logic, Syntax and Semantics of CTL, Equivalence between CTL Formulas , Model Checking Algorithm , Software Verification , Verification of real time systems hardware	**CO-3 BTL-3**
MODULE 4: Testing : Introduction to Digital Testing and Embedded System Hardware Testing	**(9)**
Introduction to Digital VLSI Testing , Automatic Test Pattern Generation (ATPG), Scan Chain based Sequential Circuit Testing, Software-Hardware Co-validation Fault Models and High Level Testing for Complex Embedded Systems ,Testing for embedded cores ,Bus and Memory Testing	**CO-4 BTL-3**
MODULE 5: Testing : Advances in Embedded System Hardware Testing and Testing for Embedded Software Systems	**(9)**
Testing for advanced faults in Real time Embedded Systems , BIST for Embedded Systems, Concurrent Testing for Fault tolerant Embedded Systems, Interaction Testing between Hardware and Software , Software testing for Reprogrammable hardware	**CO-5 BTL-3**

TEXT BOOKS	
1.	Bushnell and Agrawal, (2005), Essentials *of Electronic Testing for Digital, Memory & Mixed-Signal Circuits*, Kluwer Academic Publishers, pp. 1-690.
REFERENCE BOOKS	
1.	S. R Sabapathi, (2017) *Test engineering for Electronics Hardware*- Qmax Publications, 1st Edition, pp. 1-94
2.	M. Huth and M. Ryan,(2004), *Logic in Computer Science modeling and reasoning about systems*, Cambridge University Press, 2nd Edition, pp.1-443
3.	Peter Marwedel, P. Marwedel, (2011), *Embedded System Design*, Springer, pp. 1-237
MOOC	
1.	https://nptel.ac.in/courses/106/103/106103182/

COURSE TITLE	EMBEDDED SOFTWARE TESTING			CREDITS	3
COURSE CODE	ECH4462	COURSE CATEGORY	Honors	L-T-P-S	3-0-0-0
Version	1.0	Approval Details		LEARNING LEVEL	BTL-3

ASSESSMENT SCHEME

First Periodical Assessment	Second Periodical Assessment	Seminar/ Assignments/ Project	Surprise Test / Quiz	Attendance	ESE
15%	15%	10%	5%	5%	50%

Course Description	The course covers the fundamentals of Embedded software testing and life cycle. It also covers about dynamic, model based and coverage testing. The test management and testing for real-time cases is also studied.
Course Objective	1. To study the concepts of Testing and the Software testing life cycle. 2. To learn the dynamic Testing tools. 3. To familiarize with the Code Reviews and Static analysis tools. 4. To learn the techniques of Software Integration 5. To study the concept of Configuration Management.
Course Outcome	Upon completion of this course, the students will be able to 1. Summarize the concepts related to embedded software testing. 2. Enumerate Embedded software test life cycle, V-model and comprehend the different types of testing methods 3. Elucidate static analysis and metrics in software testing 4. Interpret top-down, bottom-up integration and testing from use cases 5. List the steps involved in test management and configuration management

Prerequisites:

CO, PO AND PSO MAPPING

CO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2
CO-1	3	-	2	-	3	-	-	1	-	1	-	1	2	3
CO-2	3	-	2	-	3	-	-	1	-	2	-	2	2	3
CO-3	3	1	2	1	3	1	-	1	-	-	-	2	2	3
CO-4	3	-	2	2	3	-	-	1	-	-	-	2	2	3
CO-5	3	-	2	-	3	-	-	1	1	-	2	2	2	3

1: Weakly related, 2: Moderately related and 3: Strongly related	
MODULE 1: FUNDAMENTALS OF EMBEDDED SOFTWARE TESTING	**(9)**
Introduction**,** Concepts of Testing**,** TEmb method, Test cases and test procedures, Principles of embedded software testing, creating a test harness, Commercial test tools. Software testing life cycle: multiple V-model, nested multiple V-model, master test planning, activities, testing by developers, testing by independent test team	**CO-1 BTL-3**
MODULE 2: TESTING METHODS	**(9)**
Dynamic Testing: Structured basis testing, Equivalence Partitions, Boundary Value Analysis, Problems with polymorphic code, Model-Based Testing: Synthesis- versus Analysis models, Generating tests from state diagrams Coverage Testing: White-box, grey and black box tests, Coverage measures – Statement, Branch, Condition, Path and others, Coverage testing tools	**CO-2 BTL-3**
MODULE 3: STATIC ANALYSIS AND CODE REVIEWS	**(9)**
Code Reviews: Benefits of reviews, Review process, Checklists Static Analysis: Static analysis concepts, the use of the compiler for static analysis, Static analysis tools, coding standards Metrics: need for metrics, Using metrics to manage and control testing, Metrics for test	**CO-3 BTL-3**
MODULE 4: SOFTWARE INTEGRATION	**(9)**
Software Integration: Importance of planning your integration, Top-down vs Bottom-up Integration, Practical integration models Testing from Use Cases: Introduction to use cases, calculating test cases, Structured Basis testing for use cases, Generating test cases from use cases Regression Testing: Purpose of regression tests, the build process	**CO-4 BTL-3**
MODULE 5: Test management	**(9)**
Configuration Management: Configuration items, Version control, Change Management, CM tools Test Management: The test process, how the test process relates to the software V-model, “Design by contract”, Test-driven development, agile development processes	**CO-5 BTL-3**
TEXT BOOKS	
1.	Paul Ammann, Jeff Offutt, (2008), *Introduction To Software Testing*, Cambridge University Press, 1st edition, pp. 1-326
MOOC	
1.	https://nptel.ac.in/courses/117/106/117106112/