Syllabus of V Semester MCA programme, w.e.f. 2018-19
(According new regulations w.e.f. 2016-17)

V SEMSTER MCA w.e.f.2018-19								
Course	Subject Name	Teaching Hrs per week	Practical Hrs/week	Examination				Credits
				Duration (Hrs.)	Marks			
					Theory/ Practical	IA	Total	
17MCACS 5.1	C# and .NET Programming	4	--	3	80	20	100	4
17MCACS 5.2	System Simulation	4	--	3	80	20	100	4
17MCACS 5.3	Data Mining	4	--	3	80	20	100	4
17MCACS 5.4	Software Testing	4	--	3	80	20	100	4
17MCACE 5.5	a. Cyber Forensic & Security b. Internet of Things c. Number Theory and Cryptography d. Digital Image Processing	4	--	3	80	20	100	4
17MCAPL 5.6	Programming Lab. –I C# Lab.	--	6	3	80	20	100	3
17MCAPL 5.7	Programming Lab. –II: s/w testing and simulation lab.	--	6	3	80	20	100	3
	Total	**20**	**12**				**700**	**26**

CS: Core Course/Subject CE: Core Elective Course OE: Open Elective

17MCACS 5.1	C# and .NET programming	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I **12Hrs**

Overview of Dynamic Web page, introduction & features of ASP.NET, Understanding ASP.NET Controls, Applications, Web servers, installation of IIS. Web forms, web form controls -server controls, client controls. Adding controls to a web form, Buttons, Text Box , Labels, Checkbox, Radio Buttons, List Box. Adding controls at runtime. Running a web Application, creating a multiform web project. Form Validation: Client side validation, server Side validation, Validation Controls : Required Field Comparison Range. Calendar control, Ad rotator Control, Internet Explorer Control.

UNIT II **10Hrs**

Overview of ADO.NET, from ADO to ADO.NET. ADO.NET architecture, Accessing Data using Data Adapters and Datasets, using Command & Data Reader, binding data to data bind Controls, displaying data in data grid. XML in .NET , XML basics, attributes, fundamental XML classes: Document, textwriter, textreader. XML validations, XML in ADO.NET, TheXMLData Ducument.

UNIT III **10Hrs**

Web services: Introduction, State management- View state, Session state, Application state. SOAP, web service description language, building & consuming a web service. Web Application deployment. Caching.

UNIT IV **10Hrs**

Threading Concepts, Creating Threads in .NET, managing threads, Thread Synchronization Security features of .NET, Role based security & Code access security, permissions

UNIT V **10Hrs**

C# and .NET, similarities & differences from JAVA, structure of C# program. Language features: Type system, boxing and un boxing, flow controls, classes, interfaces, Serialization and Persistence, Serializing an Object, Deserializing an Object Delegates, Reflection.
VB and .NET, VB .NET features.

References:

1. Mathew Macdonald, The Complete Reference ASP.NET, TMH
2. Professional ASP.NET, Wrox Publication
3. Andrew Troelsen, Pro C# with .NET 3.0, Special Edition, Dream tech Press, India
4. Steven Holzner, VB.NET Programming Black Book, Dreamtech Publications.
5. Thuan L.Thai, Hoang Lam ".Net Framework essentials: Introduction to .NET framework", Third Edition, O'Reilly & Associates Publication
6. Jesse Liberty "Learning C#" , O'Reilly & Associates Publication
7. Matt Telles, C# programming Black Book, Dreamtech Publications.
8. ASP.NET Unleashed, BPB publication

17MCACS 5.2	System Simulation	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I **10Hrs**

Introduction: Definition of system and simulation, Merits and demerits of simulation, Areas of application, Types of systems, various types of models to represent them, Discrete and Continuous systems, Steps in simulation study, Simulation Examples, Concepts of system Clocks, Event scheduling Vs Time advance algorithms.

UNIT II **10Hrs**

Statistical Models in Simulation: Random variables, discrete distributions- Binomial, Poisson and Geometric distributions, continuous distributions-Normal and Exponential distributions, Inverse transformation techniques, convolution method, Acceptance-Rejection technique, queuing models, random number generation, test for random numbers.

UNIT III **10Hrs**

Simulation Software: Selection of simulation software, simulation in C++, Simulation in GPSS, experimentation and statistical analysis tools, trends in simulation software.

UNIT IV **12Hrs**

Input Modelling: Data collection, Distribution functions such as Normal, Poisson, exponential Distributions, Goodness of fit tests, Chi square test. Input models without data, multivariate and time series input models.

Verification and Validation of Models: Guidelines for verification of models, their calibration and Validation, Face validity, Validation of model assumptions, validate input-output transformations, Use of historical Data.

UNIT V **10Hrs**

Evaluation of Simulation Experiments: Length of simulation run, static and dynamic stochastic simulations, elimination of transients, Auto correlated observations, variance reduction techniques.

References:

1. Gordan G, System Simulation, Prentice Hall of India.
2.Kishore. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, PHI.
3. John E. Freund's, Mathematical Statistics, 7/e, PHI.
4. Narsingh Deo, System simulations with Digital computers, PHI.
5. James A Payne Introduction to Simulation: Programming Techniques & Methods of Analysis McGraw Hill
6. Law A M and Ketton W D, Simulation Modeling and Analysis, McGraw Hill.
7. Deon N, System Simulation And Digital Computer , Prentice Hall of India.

17MCACS 5.3	Data Mining	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I **08Hrs**

Introduction to Data Mining: Related technologies - Machine Learning, DBMS, OLAP, Statistics, Data Mining Goals, Stages of the Data Mining Process, Data Mining Techniques, Knowledge Representation Methods, Applications, Example: weather data.

UNIT II **12Hrs**

Data Warehouse and OLAP: Data Warehouse and DBMS, Multidimensional data model, OLAP operations, Example: loan data set , Data pre-processing: Data cleaning, Data transformation, Data reduction, Discretization and generating concept hierarchies, Introduction to Weka Data Mining System, Example experiments with Weka - filters, discretization. Data mining knowledge representation: Visualization techniques, Experiments with Weka – visualization.

UNIT III **10Hrs**

Attribute-oriented analysis: Attribute generalization, Attribute relevance, Class comparison, Statistical measures, Experiments with Weka - using filters and statistics. Data mining algorithms -Association rules: Motivation and terminology, Example: mining weather data, Basic idea: item sets, Generating item sets and rules efficiently, Correlation analysis, Experiments with Weka - mining association rules.

UNIT IV **12Hrs**

Data mining algorithms- Classification: Basic learning/mining tasks, Inferring rudimentary rules: 1R algorithm, Decision trees, covering rules, Experiments with Weka - decision trees, rules. Data mining algorithms- Prediction: The prediction task, Statistical (Bayesian) classification, Bayesian networks, Instance-based methods (nearest neighbour), linear models.

UNIT V **10Hrs**

Evaluating what's been learned: Basic issues, Training and testing, Estimating classifier accuracy (holdout, cross-validation, leave-one-out), Combining multiple models (bagging, boosting, stacking), Minimum Description Length Principle (MLD), Experiments with Weka - training and testing
Clustering: Basic issues in clustering, First conceptual clustering system: Cluster/2, Partitioning methods: k-means, expectation maximization (EM), Hierarchical methods: distance-based agglomerative and divisible clustering, Conceptual clustering: Cobweb, Experiments with Weka - k-means, EM, Cobweb

References:

1. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), Morgan Kaufmann,
2. Jiawei han, Micheline Kamber,Jian Pei, Data mining concepts and techniques, 3/e, Elsevier.
3. Margaret H. Dunham, Data Mining-Introductory and Advanced Topics, Pearson Education
4. K.P.Soman, ShyamDiwakar, and V. Ajay, Insight into Data Mining: Theory and Practice, Prentice Hall of India, 2006

17MCACS 5.4	Software Testing	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I **12Hrs**

Basics of Software Testing and Examples: Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Levels of testing. Examples: Generalized pseudo code, the triangle problem, The NextDate function, The commission problem, The SATM (Simple Automatic Teller Machine) problem. Decision Table-Based Testing: Decision tables, Test cases for the triangle problem, Test cases for the NextDate function, Test cases for the commission problem, Guidelines and observations. Data Flow Testing: Definition-Use testing, Slice-based testing, Guidelines and observations.

UNIT II **10Hrs**

Levels of Testing: Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing. Integration Testing: A closer look at the SATM system, Decomposition-based, call graph- based, Path-based integrations, Case study. System Testing: Threads, Basic concepts for requirements specification, Finding threads, Structural strategies and functional strategies for thread testing, SATM test threads, System testing guidelines, ASF (Atomic System Functions) testing example.

UNIT III **10Hrs**

Interaction Testing: Context of interaction, A taxonomy of interactions, Interaction, composition, and determinism, Client/Server Testing. Issues in Object-Oriented Testing: Units for object-oriented testing, Implications of composition and encapsulation, inheritance, and polymorphism, Levels of object-oriented testing, GUI testing, Dataflow testing for object-oriented software, Examples. Class Testing: Methods as units, Classes as units.

UNIT IV **10Hrs**

Object-Oriented Integration Testing: UML support for integration testing, MM-paths for object-oriented software, A framework for object-oriented dataflow integration testing. GUI Testing: The currency conversion program, Unit testing, Integration Testing and System testing for the currency conversion program. Object-Oriented System Testing: Currency converter UML description, UML-based system testing, Statechart-based system testing. Exploratory Testing: The context-driven school, Exploring exploratory testing, Exploring a familiar example, Exploratory and context-driven testing observations.

UNIT V **10Hrs**

Model-Based Testing: Testing based on models, Appropriate models, Use case-based testing, Commercial tool support for model-based testing. Test-Driven Development: Test-then-code cycles, automated test execution, Java and JUnit example, Remaining questions, Pros, cons, and open questions of TDD, Retrospective on MDD versus TDD. A Closer Look at All Pairs Testing: The all-pairs technique, A closer look at NIST study, Appropriate applications for all pairs testing, Recommendations for all pairs testing. Software Testing Excellence: Craftsmanship, Best practice of software testing, Top 10 best practices for software testing excellence, Mapping best practices to diverse projects.

References:

1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications, 2012.
2. Aditya P Mathur: Foundations of Software Testing, Pearson, 2008.
3. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, 1st edition, John Wiley & Sons, 2011.
4. Srinivasan Desikan, Gopalaswamy Ramesh: Software testing Principles and Practices, 1st Edition, Pearson, 2012.
5. Brian Marrick: The Craft of Software Testing, 1st edition, Pearson, 2012

17MCACE 5.5	a. Cyber Forensic & Security	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I **12Hrs**

Computer Forensics Fundamentals: Computer Forensics-Law Enforcement, Computer Forensics Assistance to Human Resources/Employment Proceedings, Computer Forensics Services, Professional Forensics Methodology, Computer Forensics Technology: Military Computer Forensic Technology, Law Enforcement - Computer Forensic Technology, Business Computer Forensic Technology Computer Forensics Evidence and Capture: Data Back-up and Recovery-The Role of Back-up in Data Recovery - The Data- Recovery Solution.

UNIT II **10Hrs**

Evidence Collection and Data Seizure: Types and Rules of Evidence - Artifacts - The Chain of Custody Duplication and Preservation of Digital Evidence , Computer Evidence Processing Steps , Legal Aspects of Collecting and Preserving Computer Forensic Evidence Computer Image Verification and Authentication: Special Needs of Evidential Authentication - Practical Consideration -Practical Implementation.

UNIT III **10Hrs**

Computer Forensics analysis and validation: Addressing data-hiding techniques, Acquisitions Network Forensics: Processing Crime and Incident Scenes: Identifying digital evidence. Collecting evidence in private-sector incident scenes

UNIT IV **10Hrs**

Current Computer Forensic tools: evaluating computer forensic tool, computer forensics software tools, computer forensics hardware tools, validating and testing forensics software E-Mail Investigations, Cell phone and mobile device forensics.

UNIT V **10Hrs**

System Investigation and Ethical issues: Data Analysis Techniques- Investigating live systems (Windows & Unix), Investigating hacker tools, Ethical issues-Cybercrime

References:

1. Computer Forensics, Computer Crime Investigation by Jhon R. Vacca, Firewall Media, New Delhi.
2. Computer Forensics and Investigations by Nelson. Phillips Enfinger. Steuart, CENGAGE Learning
3. Eoghan Casey -Digital Evidence and Computer Crime, Edition 3, Academic Press, 2011
4. Principles of Information Security- Michael E. Whitman, Herbert J. Mattord, Cengage Learning, Fourth edition, 2011
5. Computer Security basics- Rick Lehtinen, O'Reilly, 2nd edition, 2006
6. Information Security Management Principles- Andy Taylor, David Alexander, Amanda Finch, David Sutton, BCS publishers, 2008.

17MCACE 5.5	b. Internet of Things	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I **08Hrs**

Fundamentals of IoT: Introduction-Characteristics-Physical design - Protocols – Logical design – Enabling technologies – IoT Levels – Domain Specific IoTs – IoT vs M2M.

UNIT II **10Hrs**

IoT Design Methodology: IoT systems management – IoT Design Methodology – Specifications Integration and Application Development.

UNIT III **12Hrs**

Building IoT With Raspberry PI: Physical device – Raspberry Pi Interfaces – Programming – APIs / Packages – Web services –

UNIT IV **10Hrs**

Building IoT with GALILEO/ARDUINO: Intel Galileo Gen2 with Arduino- Interfaces - Arduino IDE – Programming - APIs and Hacks

UNIT V **12Hrs**

Case Studies and Advanced Topics: Various Real time applications of IoT- Connecting IoT to cloud – Cloud Storage for IoT – Data Analytics for IoT – Software & Management Tools for IoT

References:

1. ArshdeepBahga, Vijay Madisetti, “Internet of Things – A hands-on approach”, Universities Press, 2015.
2. Manoel Carlos Ramon, “Intel® Galileo and Intel® Galileo Gen 2: API Features and Arduino Projects for Linux Programmers”, Apress, 2014.
3. Marco Schwartz, “Internet of Things with the Arduino Yun”, Packt Publishing, 2014

17MCACE 5.5	c. Number Theory and Cryptography	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I **10Hrs**

Elementary Number Theory: Divisibility, Division Algorithm, Euclidean Algorithm, Congruence's, Complete Residue systems, Reduced Residue systems, Fermat's little theorem, Euler's Generalization, Wilson's Theorem, Chinese Remainder Theorem, Generalized Chinese Remainder Theorem, Euler Phi-function, multiplicative property.

UNIT II **12Hrs**

Finite Fields & Quadratic Residues: Finite Fields, Primitive Roots, Quadratic Residues, Legendre Symbol, Jacobi Symbol, Gauss's lemma, Quadratic Reciprocity Law

UNIT III **10Hrs**

Primality Testing and Factorization: Primality Tests, Pseudo primes, Carmichael Numbers, Fermat's pseudo primes, Euler pseudo primes, Factorization by Pollard's Rho method, Simple Continued Fraction, simple infinite continued fractions, Approximation to irrational numbers using continued fractions, Continued Fraction method for factorization

UNIT IV **10Hrs**

Public Key Cryptosystems: Traditional Cryptosystem, limitations, Public Key Cryptography, Discrete Logarithm problem, RSA cryptosystem, Knapsack problem, Zero knowledge protocols and oblivious transfer.

UNIT V **10Hrs**

Elliptic Curve Cryptography: Cubic Curves, Singular points, Discriminant, Introduction to Elliptic Curves, Geometry of elliptic curves over, Group structure, Points of finite order, Elliptic Curves over finite fields, Discrete Log problem for Elliptic Curves, Elliptic Curve Cryptography, Factorization using Elliptic Curve, Lenstra's algorithm, ElGamal Public Key Cryptosystem for elliptic curves.

References:

1. An Introduction to Cryptography, R.A. Mollin , Second Edition (Chapman & Hall, 2006)
2. Rational Points on Elliptic Curves, Silverman and Tate
3. Guide to elliptic curve cryptography Hankerson, Menezes, Vanstone
4. Introduction to Algorithms: T. H. Cormen, C. E. Leiserson, R. Rivest and C., Stein Prentice Hall India, 2nd Edition.
5. A Course in Number Theory and Cryptography: Neal Koblitz, Springer-Verlag, New York Inc.
6. Cryptography and Network security: Principles and Practice, William Stallings, Pearson Education
7. Introduction to Cryptography with Coding Theory, Second Edition, W.Trappe and L. C. Washington, Pearson Education 2007.
8. Cryptography: Theory and Practice, Douglas R. Stinson, CRC Press.
9. Randomized Algorithms, R. Motwani and P. Raghavan, Cambridge University Press

17MCACE 5.5	d. Digital Image Processing	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I **10Hrs**

Digital Image Fundamentals: The origins of Digital Image Processing, Examples of Fields that Use Digital Image Processing, Fundamentals Steps in Image Processing, Elements of Digital Image Processing Systems, Image Sampling and Quantization, Basic relationships between pixels.

UNIT II **12Hrs**

Image Enhancement in the Spatial Domain: Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic and Logic operations, Basics of Spatial Filters, Smoothening and Sharpening Spatial Filters, Combining Spatial Enhancement Methods.

UNIT III **10Hrs**

Image Enhancement in the Frequency Domain: Introduction to Fourier Transform and the frequency Domain, Smoothing and Sharpening Frequency Domain Filters, Homomorphic Filtering. Colour Image Processing: Colour models, pseudocolor image processing, colour transformations, smoothening and sharpening.

UNIT IV **10Hrs**

Image Restoration: A model of The Image Degradation / Restoration Process, Noise Models, Restoration in the presence of Noise Only Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering, Linear Position-Invariant Degradations, Estimation of Degradation Function, Inverse filtering, Wiener filtering, Constrained Least Square Filtering, Geometric Mean Filter, Geometric Transformations.

UNIT V **10Hrs**

Image Segmentation: Detection of Discontinuities, Edge linking and boundary detection, Thresholding, Region Oriented Segmentation, Motion based segmentation.
Morphological Processing: Some basic Morphological operations.

References:

1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 3/e, Pearson Education.
2. Anil K. Jain, Fundamentals of Digital Image Processing', Pearson .
3. Kenneth R. Castleman, Digital Image Processing, Pearson.
4. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins,Digital Image Processing using MATLAB,P earson Education, Inc..
5. William K. Pratt, , Digital Image Processing, John Wiley, New York,
6. Milan Sonka et al, Image Processing, Analysis And Machine Vision, Brookes/Cole, Vikas Publishing House.

17MCAPL 5.6	Programming Lab. –I C# Lab.	
Credits: 3	Teaching: 6Hrs/week	Max. Marks: 80 IA: 20

Suggested Assignments

1. Describe the enumerations programming constructs, which provides a human-readable form of a series of related constant values in C#.
2. Create classes, they are reference types in C# and hence are allocated on the heap. Classes provide object-oriented constructs such as encapsulation, polymorphism, and inheritance. For instance, the program should print John. Doe twice, illustrating that objects are reference types, allocated on the heap implement the same using C#.
3. Check Whether the Entered Year is a Leap Year or Not
4. Describe Arrays and Strings methods with suitable C# program
5. Program to display the addition, subtraction, multiplication and division of two number using console applications.
6. Program to display the first 10 natural numbers and their sum using console application.
7. Program to display the addition using the windows application.
8. Work with Page using ASP.Net.
9. Write a program to convert input string from lower to upper and upper to lower case.
10. Work with forms using ASP.NET.
11. Describe access data source through ADO.NET.
12. Perform operator overloading.
13. Describe delegates, events, errors and exceptions
14. Find the second largest element in a single dimensional array.
15. Program to illustrate the use of different properties in C#.
16. Demonstrate Command line arguments processing.
17. Program to multiply to matrices using Rectangular arrays.
18. Demonstrate Use of Virtual and override keyword in C# with a simple Program.

17MCAPL 5.7	Programming Lab. –II: Software testing and Simulation lab.	
Credits: 3	Teaching: 6Hrs/week	Max. Marks: 80 IA: 20

Suggested Assignments

1. Students are given a program that draws a sequence of differently coloured rectangles and are asked to modify the code so that the result will be a sequence of rectangles that gradually change in colour from the colour of the first to the colour of the last. The algorithm to blend the correct colour for each rectangle requires the students to use proportions based on the distance each rectangle is from the first and last rectangles.

2. The spring and fall change between Standard and Daylight savings time creates an interesting problem for telephone bills. Focus your thinking on the complications arising from the daylight saving time transitions. Create a table that shows risks, equivalence classes, boundary cases, and expected results for a long distance telephone service that bills calls at a flat rate of Rs.5.05 per minute. Assume that a chargeable time of a call begins when the called party answers, and ends when the calling party disconnects.

3. Distinguish between using code coverage to highlight what has not been tested from using code coverage to measure what has been tested. Describe some benefits and some risks of each type of use. (In total, across the two uses, describe three benefits and three risks).
   List and describe five different dimensions (different "goodness's") of "goodness of tests".

4. A program asks you to enter a password, and then asks you to enter it again. The program compares the two entries and either accepts the password (if they match) or rejects it (if they don't). you can enter letters or digits. How many valid entries could you test? (Please show and/or explain your calculations).

5. A program is structured as follows:
   - It starts with a loop; the index variable can run from 0 to 20. The program can exit the loop normally at any value of the index.
   - Coming out of the loop, there is a case statement that will branch to one of 10 places depending on the value of X. X is a positive, non-zero integer. It can have any value from 1 to MaxInt.
   - In 9 of the 10 cases, the program executes X statements and then goes into another loop. If X is even, the program can exit the loop normally at any value of its index, from 1 to X. If X is odd, the program goes through the loop 666 times and then exits. In the 10th case, the program exits.

     Ignore the possibility of invalid values of the index variable on X. How many paths are there through this program? Please show and/or explain your calculations.

6. Consider a program with two loops, controlled by index variables. The first variables increments (by 1 each iteration) from -3 to 20. The second variables increments (by 2 each iteration)from 10 to 20.The program can exist from either loop normally at any value of the loop index .(Ignore the possibility of invalid values of the loop index.)
   - If these were the only control structures in the program, how many paths are there through the program?
     I. If the loops are nested
     II. If the loops are in series, one after the other
   - If you could control the values of the index variables, what test cases would you run if you were using a domain testing approach?
   - Please explain your answers with enough detail that I can understand how you arrived at the numbers.

7. Imagine testing a date field. The field is of the form MM/DD/YYYY (two digit month, two digit day, 4 digit year). Does equivalence class analysis and boundary tests that you would run in order to test the field. (Don't bother with non-numeric values for these fields).

8. Imagine testing a file name field. For example, go to an open file dialog, you can enter something into the field. Do a domain testing analysis: List a risk, equivalence classes appropriate to the risk, and best representatives of the equivalence classes. For each test case(use a best representative), briefly explain why this is a best representative. Keep doing this until you have listed 12 best-representative test cases.

9. Take any system (e.g. ATM system) and study its system specifications and report the various bugs.
    a. Machine is accepting ATM card.
    b. Machine is rejecting expired card.
    c. Successful entry of PIN number.
    d. Unsuccessful operation due to enter wrong PIN number 3 times.
    e. Successful selection of language
    f. Successful selection of account type

10. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary value analysis, execute the test cases and discuss the results.

11. Design, develop, code and run the program in any suitable language to implement the Next Date function. Analyze it from the perspective of equivalence class value testing, derive different test cases, execute these test cases and discuss the test results.

12. Write the test cases for any known application (e.g. Banking application)
    a. Checking mandatory input parameters
    b. Checking optional input parameters
    c. Check whether able to create account entity.
    d. Check whether you are able to deposit an amount in the newly created account (and thus updating the balance)
    e. Check whether you are able to withdraw an amount in the newly created account (after deposit) (and thus updating the balance)

13. Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects)

14. Program test a to update 10 student records into table into Excel file

15. Program test a to provide total number of objects present / available on the page.

16. Program test to get the number of list items in a list / combo box.

17. Program to test a count number of items present on a desktop

18. Simulate of the M/G/1 queue using Lindley's equation.

19. Simulate the SAN (Stochastic Activity Network) as the maximum path through the network.

20. Simulation of paging Technique in operating System.

21. Simulate a simple Calculator.

Semester End Examination Question Paper Pattern

Max Marks: 80 Duration - 3 Hours.

Theory Question Paper Pattern

- There shall be eight questions of 16 marks each.
- Each question may have sub questions (a),(b) / (a),(b),(c)
- There shall be a minimum of one question from each unit
- There shall be not more than 2 questions from any unit.
- The student has to answer any five full questions for scoring full marks

Internal Assessment Scheme

Internal Assessment– 20 Marks:

IA Test : 14 marks
Attendance : 03 marks
Seminar/assignment : 03 marks

Two tests shall be conducted, one during the mid of the semester and another at the end of the semester of 1hour duration each.

First IA Marks : 14
Second IA Marks : 14

Average of the two tests shall be taken as final marks.

Attendance Marks allocation scheme

Attendance (in percentage)	90 and above	80 and above but below 90	75 and above but below 80	below 75
Marks	3	2	1	no Marks*

*not eligible for appearing semester end examination (as per Regulation 7.7)

Note: Guidelines notified by the University from time-to-time shall be followed for IA.

VI SEMSTER MCA w.e.f.2018-19								
Course	Subject Name	Teaching Hrs per week	Contact Hrs/ Per Week	Examination and Credits				
				Duration (Hrs.)	Marks			Credits
					Practical	IA	Total	
17MCACS 6.1	Project Work	--	32Hrs	3	250*	50	300	16
	Total	**--**	**32Hrs**				**300**	**16**

*** Project Report Evaluation: 100; Project Demo & presentation: 100 Marks; Viva-Voce: 50 Marks**

Internal Assessment Scheme

Internal Assessment– **50 Marks:**

Two tests shall be conducted, one during the mid of the semester and another at the end of the semester of 1hour duration each.

First IA Marks : 25
Second IA Marks : 25