Name of Program : MCA
(For Regular and Lateral Entry Students)

COURSE CODE	CATEGORY	COURSE NAME	L	T	P	CREDITS	THEORY END SEM University Exam	THEORY Two Term Exam	THEORY Teachers Assessment*	PRACTICAL END SEM University Exam	PRACTICAL Teachers Assessment*
MCCA301	COMPULSORY	Design Methods & Analysis of Algorithms	3	1	0	4	60	20	20	0	0

Legends: L - Lecture; T - Tutorial/Teacher Guided Student Activity; P – Practical;   C - Credit;

Q/A – Quiz/Assignment/Attendance, MST - Mid Sem Test.

*Teacher Assessment shall be based on following components: Quiz / Assignment / Project /Participation in class (Given that no component shall be exceed 10 Marks)

Course Educational Objectives (CEOs):
- To familiarize the students with the need and scope of the subject.
- To develop the analytical skills of students so that they can better understand the problem and devise algorithms efficiently.
- Using simple and well drawn illustrations develop their analytical and programming skills.
- To cover the various data structures and their applications so that aspirants can explore this territory
- to take on the more challenging concepts.

Course Outcomes (Cos): After the completion of the course the student will be able to
- Understand the real world problems and model them
- Understand the algorithms and the algorithm design process
- Choose a suitable strategy to devise solution of a given problem
- identify, formulate and solve programming problems
- select appropriate data structures for the solution of a given problem
- analyze the algorithms for correctness and in terms of complexity for best, worst and average cases
- function on multi-disciplinary teams
- understand the professional and ethical responsibility

Pre-requisites: Data structure and Discrete structures, models of computation.

Unit-I

Introduction to Algorithm: Definition, Criteria of Algorithm, Algorithm Analysis, Time and Space complexity, Asymptotic notation: Big Oh, Omega and Theta, Best, Average and Worst

case analysis.

Unit-II
Design of Algorithm, Types of algorithm strategies, Recurrence relation: Master method, Substitution method, Analysis of algorithm, Brute-force approach: Sequential Search, Selection Sort.
Divide and Conquer: Structure of divide-and-conquer algorithms: examples; Binary search, Merge sort, Quick sort, Strassen's Multiplication; Analysis of Divide and Conquer methods, Comparison of Searching and Sorting Algorithms.

Unit-III
Graph searching and Traversal: Overview, Traversal methods (Depth First and Breadth First search)
Greedy Method: Overview of the greedy paradigm examples of exact optimization solution (minimum cost spanning tree), Approximate solution (Knapsack problem), Single Source Shortest Paths.
Branch and Bound: LC searching Bounding, FIFO Branch and Bound, LC Branch and Bound application: 0/1 Knapsack Problem, Traveling Salesman Problem.

Unit-IV
Dynamic programming: An Overview, Difference between Dynamic Programming and Divide And Conquer, Applications: Shortest Path in Graph, Matrix Chain Multiplication, Traveling Salesman Problem, Longest Common Sequence.
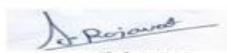**Back tracking:** Overview, 8-Queens problem and Knapsack problem

Unit-V.
Computational Complexity: Complexity measures, Polynomial Vs non-polynomial time complexity; NP-hard and NP-complete classes, Relation among P, NP, NPC and NPH. Examples. Combinational Algorithms, String Processing Algorithm, Algebraic Algorithms, Set Algorithms.

Suggested Readings:
1. Ullman, "Analysis and Design of Algorithm", TMH
2. Goodman, "Introduction to the Design & Analysis of Algorithms, TMH-2002.
3. Sara Basse, A. V. Gelder, "Computer Algorithms," Addison Wesley
4. T. H. Cormen, Leiserson, Rivest and Stein, "Introduction of Computer algorithm," PHI
5. E. Horowitz, S. Sahni, and S. Rajsekaran, "Fundamentals of Computer Algorithms," Galgotia Publication

Name of Program : MCA
(For Regular and Lateral Entry Students)

COURSE CODE	CATEGORY	COURSE NAME	L	T	P	CREDITS	TEACHING & EVALUATION SCHEME				
							THEORY			PRACTICAL	
							END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*
MCCA302	COMPULSORY	Software Engineering	3	1	4	6	60	20	20	30	20

Legends: L - Lecture; T - Tutorial/Teacher Guided Student Activity; P – Practical;   C - Credit;
Q/A – Quiz/Assignment/Attendance, MST - Mid Sem Test.
*Teacher Assessment shall be based on following components: Quiz / Assignment / Project/Participation in class (Given that no component shall be exceed 10 Marks)

<u>**Course Education Objectives (CEOs):**</u>
- To provide knowledge about system development.
- To impart knowledge about software process models.
- To provide detailed knowledge about software design.
- To acquaint students with software quality and testing.
- To provide study of MIS.

<u>**Course Outcomes (COs):**</u>Students will be having:
- An ability to understand system design and its constraints.
- An ability to apply knowledge of software engineering.
- An ability to design a system, a component or process to meet desired needs.
- An ability to identify, formulate and solve engineering Problems
- An ability to measure and to understand quality issues.
- An ability to understand MIS and DSS.

UNIT -I
Introduction to Software Engineering: Software problem, Software engineering problem, Software engineering approach, Software characteristics and Applications. The system concept, characteristics of system, elements of system, The System Development Life Cycle, The Role of System Analyst.

UNIT -II
Software Processes: Software processes and its components, characteristics of software processes, Software development processes: Linear Sequential model, Prototyping model, RAD model, Iterative Enhancement model, Spiral model, Component based development,

Comparative study of various development models. Fourth Generation Techniques.

UNIT-III
Software Project Planning : Project planning objectives, Decomposition Techniques, Empirical estimation models, The Make/Buy Decision., Risk analysis. Software Design: Design Principles, Cohesion & Coupling.

UNIT-IV
Software Quality Assurance : Quality Concepts, The Quality Movement, Software Quality Assurance, Software Reviews, Formal Technical Reviews, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, Software Testing Techniques: Testing fundamentals, White box testing, Black box testing. Testing Strategies: A strategic approach of software testing strategic issues, unit testing, integration testing, validation testing, system testing.

UNIT-V
MIS & DSS: Introduction to MIS, long range planning, development and implementation of an MIS, Applications of MIS in manufacturing sector and service sector. Decision Support System concepts, types of DSS.

Suggested Readings:
1. R. S. Pressman, "Software Engineering – A practitioner's approach", 6th ed., McGraw Hill Int.
2. PankajJalote "Software Engg" Narosa Publications.
3. Ian Sommerville : Software Engineering 6/e (Addison-Wesley)
4. Richard Fairley : Software Engineering Concepts (TMH)
5. Elis Awad, "System Analysis & Design", Galgotia publications
6. W.S. Jawadekar: Management Information Systems, TMH Publication, India

PRACTICAL LIST
1. To study the Software Development Life Cycle.
2. To understand and apply good Software Analysis and Design practices
3. To study Data Flow Diagrams (DFDs) and levels in DFDs.
4. To create Data Dictionary for some applications
5. To use various information gathering tools (Questionnaire, Interview, On Site Survey)
6. To choose suitable software development process models for developing different applications.
7. Perform Feasibility Study and to create Feasibility Report for applications.
8. To make decision whether to buy/lease/ develop the software.
9. To understand and create Use Case Diagram.
10. To study Functional Point Analysis.
11. To devise Test Cases for software testing, black-box, white-box testing and different types of testing.
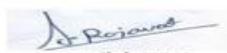
12. To study the Risk Management during the software development.
13. To assure Quality of Software, Statistical Software Quality Assurance, Reliability of Software.
14. To understand and apply concepts of Project Management
15. Case study (MIS and DSS)

Name of Program : MCA
(For Regular and Lateral Entry Students)

COURSE CODE	CATEGORY	COURSE NAME	L	T	P	CREDITS	THEORY END SEM University Exam	THEORY Two Term Exam	THEORY Teachers Assessment*	PRACTICAL END SEM University Exam	PRACTICAL Teachers Assessment*
MCCA303	COMPULSORY	E- Commerce	3	1	0	4	60	20	20	0	0

Legends: L - Lecture; T - Tutorial/Teacher Guided Student Activity; P – Practical; C - Credit; Q/A – Quiz/Assignment/Attendance, MST - Mid Sem Test.

*Teacher Assessment shall be based on following components: Quiz / Assignment / Project / Participation in class (Given that no component shall be exceed 10 Marks)

Course Educational Objectives (CEOs):
- To develop an understanding of scope of E-Commerce.
- To develop an understanding of electronic market and market place.
- To develop an understanding of business models.
- To develop an understanding of legal issues, threats of E-Commerce.

Course Outcomes (COs) : Students will be able to
- Students would be able to analyze the concept of electronic market and market place.
- Students would be able to understand the business models.
- Students would be able to understand the business standards.
- Students would be able to understand the legal and security issues.

Unit I
Introduction to e-commerce: History of e-commerce, Potential Benefits, Limitations, Classifications-business models, legal; environment of e-commerce, ethical issues, Basics of Sending & Receiving, , advantages and disadvantages of e-commerce.

Unit II

Electronic Payment Systems: Advantages and risks, Types of Payment System (Credit Cards, E-Cash, Smart-Cards). e-credit accounts, e-money, value chain and supply chain, marketing strategies, marketing on the web, advertising on the web, customer service and support, introduction to m-commerce.

Unit III

E-Commerce Applications: Entertainment, E-Marketing, E-Advertising, Search Engines, E-Government, issues in e-governance applications, evolution of e-governance, its scope and content, benefits and reasons for the introduction of e-governance, case study: e-commerce in passenger air transport.

Unit IV

Electronic Data Interchange: Non EDI System, Partial EDI System, Fully Integrated EDI System, Prerequisites for EDI. Issues of EDI: Legal issues, Security issues, Privacy issues - readiness, e-government readiness, E- Framework, step & issues, application of data warehousing and data mining in e-government.

Unit V

E-Advertising Techniques: Banners, Sponsorships, Portals, Online Coupons E-Government systems security: Challenges and approach to e-government security, security concern ine-commerce, security for server computers, communication channel security, and security for client computers. Case studies: NICNET-role of nationwide networking in e governance, e- seva.

Suggested Readings:
1. Gary P. Schneider, "E-commerce", Cengage Learning India.
2. C.S.R. Prabhu, "E-governance: concept and case study", PHI Learning Private Limited.
3. V. Rajaraman, "Essentials of E-Commerce Technology", PHI Learning Private Limited.
4. David Whiteley, "E-commerce study, technology and applications", TMH.
5. J. Satyanarayan, "E-government: The science of the possible", PHI Learning Private Limited.
6. P.T. Joseph, "E-Commerce an Indian Perspective", PHI Learning Private Limited.
7. Hanson and Kalyanam, "E-Commerce and Web Marketing", Cengage Learning India.
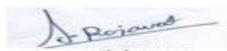
Chairperson
Board of Studies
(Computer Science & Engineering,
Information Technology & Computer Applications)
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore

Joint Registrar
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore

ShriVaishnavVidhyapeethVishvavidhyalaya, Indore

Institute of Computer Applications

Name of Program : MCA
(For Regular and Lateral Entry Students)

COURSE CODE	CATEGORY	COURSE NAME	L	T	P	CREDITS	THEORY END SEM University Exam	THEORY Two Term Exam	THEORY Teachers Assessment*	PRACTICAL END SEM University Exam	PRACTICAL Teachers Assessment*
MCCA304	COMPULSORY	Object Oriented Programming	3	1	4	6	60	20	20	30	20

Legends: L - Lecture; T - Tutorial/Teacher Guided Student Activity; P – Practical;   C - Credit;

Q/A – Quiz/Assignment/Attendance, MST - Mid Sem Test.

*Teacher Assessment shall be based on following components: Quiz/Assignment/Project/Participation in class (Given that no component shall be exceed 10 Marks)

Course Educational Objectives (CEOs):
- To familiarizes the students with Object Oriented Methodology.
- To make students well versed with C++ language so that they can solve problems efficiently.
- Using simple and well drawn illustrations develop their programming skills.
- To cover the various data structures and their applications so that aspirants can explore this territory
- to take on the more challenging concepts.

Course Outcomes (Cos):The student will be able to
- Understand different programming paradigms, Evolution of programming languages, Programming styles
- Differentiate and compare structured and object oriented approach
- Have a mental makeup of programming in large using various OOPs features
- Use the concept of reusability
- Handle files and Exceptions
- Apply the OO design and analysis concepts
- Design efficient solutions for real world problems.
- Use UML concepts and different UML diagrams to represent the systems.

UNIT-I
OOPs, Software Crises, Evolution of programming methodologies, origin of C++, comparison of OOPs with other paradigms, Features, Merits and demerits of OOPs loops and decisions, structures and functions, object and classes, object arrays, constructor and destructor functions.

Unit-II
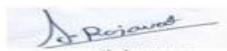
Chairperson
Board of Studies
(Computer Science & Engineering,
Information Technology & Computer Applications)
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore
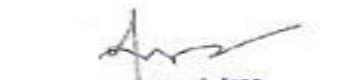
Joint Registrar
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore

Introduction to Objects and classes-Defining the class, data members and member functions, creating objects, access specifiers, Nested classes, local classes, empty class. Friend function and friend class. Objects and functions, static members, new and delete operator, pointer to object, pointer to class members, wild pointers, dangling pointers, smart pointers.

UNIT-III
Operator and function overloading, Inheritance.
Polymorphism, virtual functions, abstract base classes and pure virtual function, friend function, early and late binding.

UNIT-IV
C++ I/O system, formatted I/O, creating insertors and extractors, file I/O basis, creating disk files and file manipulations using seekg(), seekp(), tellg() and tellp() functions, exception handling: try, catch and throw. Inside an Exception Handler, defining your own exceptions

UNIT-V
UML concepts, object-oriented paradigm and visual modeling, UML diagrams, UML specifications, object model, object oriented design, identifying classes and object, object diagrams.

List of Programs:
1. Define copy constructor. Explain its significance. Under which condition is it invoked? Support your answer with an example.
2. Explain the purpose of function overloading. Write a C++ program that uses afunction to check whether a given number is divisible by another number or not. Give proper comments whether the denominator number is a prime number or not.
3. Define a friend function. Explain the relation of friend function with respect to Public, Private and Protected data members of the class.
4. Write a program in C++
   - To generate the following number pyramid

   ```
                   0
               1   0   1
           2   1   0   1   2
       3   2   1   0   1   2   3
   4   3   2   1   0   1   2   3   4
   ```

   - Generate the Fibonacci series using recursion.
   - Create two function max and min to find the largest and smallest numbers respectively. Function select having arguments as a pointer to function and numbers. User will input the choice to find out the max or min.
   - Convert a 2 digit octet number into binary number and prints binary equivalent.
5. Write a function in C++ that take one string argument and return a reversed string.
6. Write a C++ program using structure to store information of 10 employee (id_no,
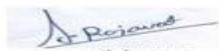
empname, empadd, sal) and display information of an employee depending upon the employee no given.

7. Create a class containing name & telephone number as two of its fields. Write a C++ Program using an object to do the following

   (a) to add a record   (b) to search of telephone number for a given name

   (c) to determine the name of telephone number is known

   (d) Updating the data file, whenever there is a change in telephone number.

8. Imagine a ticket selling booth at a fair. People passing by are requested to purchase a ticket. A ticket is priced as Rs 2.50/-. The booth keeps the track of the number of people that have visited the booth and of the total amount of money collected. Model this ticket selling booth with a class called **ticbooth** including following members: number of people visited, total number of money collected.

   Member function:

   (1)to assign initial values (2) to increment only people total in case ticket is not sold out.
   (3) to increment people total as well as amount total in case ticket is not sold out (4) to display the two totals (5) to display the number of tickets sold out.

9. Create a class Bank account of 20 customers with the following data members depositor no, depositor name, account type(S for saving & C for current), balance amount. The class also contains the member functions to do the following :

   (a) initialize the data member   **(b)** deposit money   **(c)** Display the customer information

   (d) withdraw money after checking the balance (min bal for saving account is 500 and for current   1000)   **(e)** search the depositor details according to name or number or both.

11. Write a program that invoke a function newdate() to return an object Date type. The function newdate( ) take two parameters: An object oldadate of Date type,   Number of days in integer

    Calculate the newdate as olddate + numbers of days and return newdate.

12. Write a program to interchange the private values of two classes by using a friend function swap().

13. Create a **Message** class with a constructor that takes a single **string** with a default value. Create a private member **string**, and in the constructor simply assign the argument **string** to your internal **string**. Create two overloaded member functions called **print( )**: one that takes no arguments and simply prints the message stored in the object, and one that takes a **string** argument, which it prints in addition to the internal message. Does it make sense to use this approach instead of the one used for the constructor?

14. Write C++ code to create a class date. Using this class, calculate the age of a person as on the current date.

15. Create a class contain Bookno, Book Title, Book price. The class also store and display the complete information of book. **Total_cost()** function calculate the cost of for **N** number of copies, where **N** is passed to the function as argument and **purchase**() function to ask the user it input the number of copies to be purchased. It invoke **Total_cost()** and print the total cost to be paid by the user.

Chairperson
Board of Studies
(Computer Science & Engineering,
Information Technology & Computer Applications)
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore

Join: Registrar
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore

16. Define a class string use overloaded = = operator to compare two string.
17. Find the largest value of two objects using > operator.
18. Write a C++ program to print the number between 1 to 100 and 100 to 1 using operator overloading.
19. Define a class String that could work as a user defined string type. Include a constructors that will enable user to create an uninitialized string String s1; and also to initialize an object with a string constant at the time of creation like String s2("Welcome"); Include a function that adds two strings to make a third string. Note that the statement s2=s1; will be perfectly reasonable expression to copy one string to another .
20. Write a complete program to test your class to see that it does the following tasks:
    (a) Create uninitialized string objects.   (b) Creates objects with string constants
    (c) Concatenates two string   (d) Display a desired string objects

Suggested Readings:

1. Lafore R. "Object Oriented Programming in C++", Galgotia Publication
2. Schildt, "C++ the complete reference 4ed, 2003.
4. Hans Erit Eriksson, "UML 2 toolkit" Wiley.
5. Balagurusawmy, "Object Orienter Programming with C++".
6. K.R. Venugopal, Rajkumar, T. ravishankar, ' Mastering C++', TMH
6. B.G., Boach, "Object Oriented Analysis & Design with Applications", AddisionWesly.
7. Boggs, "Mastering UML" BPB Publications
9. Lee, "UML & C++ a practical guide to Object Oriented Development 2 ed, Pearson.

Chairperson
Board of Studies
(Computer Science & Engineering,
Information Technology & Computer Applications)
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore

Joint Registrar
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore