Semester – IV (w.e.f. Session 2016-17)

S. N.	Course No.	Course Title	Teaching Schedule				Allotment of Marks				Dur of Exam (Hrs.)
			L	T	P	Hr/ Wk	Theory	Sessional	Practical	Total	
1.	AS-201 N	Mathematics - III	3	1	-	4	75	25	--	100	3
2.	IT-202 N	Computer Organization & Architecture	3	1	-	4	75	25	--	100	3
3.	IT- 204 N	Programming Languages	3	1	--	4	75	25	--	100	3
4.	IT-206 N	Object Oriented Programming Using C++	3	1	--	4	75	25	--	100	3
5.	IT-208 N	Operating Systems	3	1	--	4	75	25	--	100	3
6.	IT-210 N	Fundamentals of Microprocessor Interfacing & Application	3	1	--	4	75	25	--	100	3
7.	IT-212 N	Programming with C++ Lab	--	--	2	2	---	40	60	100	3
8.	IT-214 N	Microprocessor & Interfacing Lab	--	--	2	2	---	40	60	100	3
9.	IT-216 N	Computer Hardware and Troubleshooting Lab	--	--	2	2	---	40	60	100	3
10.	IT-218 N	Programming with MATLAB	--	-- -	2	2	--	40	60	100	3
		Total	**18**	**5**	**9**	**32**	**450**	**310**	**240**	**1000**	
11.	MPC-201 N*	Environmental Studies*	3	--	--	3	75	25	-	100	3

***MPC-201N is a mandatory course which will be non-credit subject and students has to get pass marks in order to be eligible for the award of degree.**

Note:- All the students have to undergo 4-6 week industrial training after 4th semester and it will be evaluated in 5th Sem.

AS-201 N **Mathematics-III**

Lecture	Tutorial	Practical	Major Test	Minor Test	Total Time
3	1	-	75	25	3H

Purpose To provide the conceptual knowledge of Engineering mathematics
Course Outcomes
CO 1 To study various fundamental concepts of Fourier series and Fourier Transformation.
CO 2 To study and understand the functions of a complex variables.
CO 3 To study the Probability Distributions.
CO 4 To study the linear programming problem formulation.

UNIT – I

Fourier Series : Euler's Formulae, Conditions for Fourier expansions, Fourier expansion of functions having points of discontinuity, change of interval, Odd & even functions, Half-range series.
Fourier Transforms : Fourier integrals, Fourier transforms, Fourier cosine and sine transforms.
Properties of Fourier transforms, Convolution theorem, Perseval's identity, Relation between Fourier and Laplace transforms, Fourier transforms of the derivatives of a function, Application to boundary value problems.

UNIT-II

Functions of a Complex Variables : Functions of a complex variable, Exponential function ,Trigonometric, Hyperbolic and Logarithmic functions, limit and continuity of a function, Differentiability and analyticity.
Cauchy-Riemann equations, Necessary and sufficient conditions for a function to be analytic, Polar form of the Cauchy-Riemann equations, Harmonic functions, Application to flow problems, Conformal transformation, Standard transformations (Translation, Magnification &rotation, inversion & reflection, Bilinear).

UNIT-III

Probability Distributions : Probability, Baye's theorem, Discrete & Continuous probability distributions, Moment generating function, Probability generating function, Properties and applications of Binomial, Poisson and normal distributions.

UNIT-IV

Linear Programming: Linear programming problems formulation, Solution of Linear Programming Problem using Graphical method, Simplex Method, Dual-Simplex Method.

Paper Setter's Note: 8 questions of 15 marks each distributed in four sections are to be set taking two from each unit. The candidate is required to attempt five questions in all, taking at least one from each of the four sections.

Text Book

1. Higher Engg. Mathematics : B.S. Grewal
2. Advanced Engg. Mathematics : E. Kreyzig

Reference Book

1. Complex variables and Applications : R.V. Churchil; Mc. Graw Hill
2. Engg. Mathematics Vol. II: S.S. Sastry; Prentice Hall of India.
3. Operation Research : H.A. Taha.
4. Probability and Statistics for Engineer : Johnson. PHI.

IT-202 N	Computer Organization & Architecture					
Lecture	Tutorial	Practical	Major Test	Minor Test	Total	Time
3	1	-	75	25	100	3
Purpose	To familiarize with the architecture of computer system					
COURSE OUTCOMES						
CO1	To study the architecture and instruction set.					
CO2	To know about how instruction can be fetch and execute.					
CO3	To study about the memory.					
CO4	To study the parallelism and interrupts.					

Unit-1

General System Architecture: Von-neumann Model, Store program control concept, Flynn's classification of computers (SISD, MISD, MIMD); Multilevel viewpoint of a machine: digital logic, micro architecture, ISA, operating systems, high level language; structured organization; CPU, caches, main memory, secondary memory units & I/O; Performance metrics; MIPS, MFLOPS, Common us system

Instruction Set Architecture: Instruction set based classification of processors (RISC, CISC, and their comparison); addressing modes: register, immediate, direct, indirect, indexed; Operations in the instruction set; Arithmetic and Logical, Data Transfer, Machine Control Flow; Instruction set formats (fixed, variable, hybrid).

Unit-2

Basic non pipelined CPU Architecture: CPU Architecture types (accumulator, register, stack, memory/ register) detailed data path of a typical register based CPU, Fetch-Decode-Execute cycle (typically 3 to 5 stage); microinstruction sequencing, implementation of control unit, Enhancing performance with pipelining.

Unit-3

Memory Hierarchy & I/O Techniques: The need for a memory hierarchy (Locality of reference principle, Memory hierarchy in practice: Cache, main memory and secondary memory, Memory parameters: access/ cycle time, cost per bit); Main memory (Semiconductor RAM & ROM organization, memory expansion, Static & dynamic memory types); Cache memory (Associative & direct mapped cache organizations. Allocation & replacement polices, segments, pages & file organization, virtual memory.

Unit-4

Introduction to Parallelism: Goals of parallelism (Exploitation of concurrency, throughput enhancement); Amdahl's law; Instruction level parallelism (pipelining, super scaling –basic features); Processor level parallelism (Multiprocessor systems overview).Types of interrupts; Memory Hierarchy. Programmed I/O, DMA & Interrupts.

Text Books:

- Computer Organization and Design, 2nd Ed., by David A. Patterson and John L. Hennessy, Morgan 1997, Kauffmann.
- Computer Architecture and Organization, 3rd Edi, by John P. Hayes, 1998, TMH.

Reference Books:

- Operating Systems Internals and Design Principles by William Stallings,4th edition, 2001, Prentice-Hall Upper Saddle River, New Jersey
- Computer Organization, 5th Edi, by Carl Hamacher, Zvonko Vranesic,2002, Safwat Zaky.
- Structured Computer Organisation by A.S. Tanenbaum, 4th edition, Prentice-Hall of India, 1999, Eastern Economic Edition.
- Computer Organisation & Architecture: Designing for performance by W. Stallings, 4th edition, 1996, Prentice-Hall International edition.
- Computer Architecture & Organisation by M. Mano, 1990, Prentice-Hall.
- Computer Architecture- Nicholas Carter, 2002, T.M.H.

Paper Setter's Note: 8 questions of 15 marks each distributed in four sections are to be set taking two from each unit. The candidate is required to attempt five questions in all, taking at least one from each of the four sections.

IT – 204 N	Programming Languages					
Lecture	Tutorial	Practical	Major Test	Minor Test	Total	Time
3	1	-	75	25	100	3 Hour
Purpose	To introduce the principles and paradigms of programming languages for design and implement the software intensive systems.					
Course Outcomes						
CO 1	To study the syntax and semantics of programming language.					
CO 2	To know about the data type concept.					
CO 3	To study the control statement of programming language					
CO 4	To know about the storage management					

Unit-I:

Introduction, Syntax and Semantics Introduction: A brief history, characteristics of a good programming language, Programming language translators compiler & interpreters , Elementary data types –data objects, variable & constants, data types, Specification & implementation of elementary data types, Declarations, type checking & type conversions, Assignment & initialization, Numeric data types, enumerations, Booleans & characters. Syntax& Semantics: Introduction, general problem of describing syntax, formal method of describing syntax, attribute grammar .

Unit-II:

Structured data objects, Subprograms and Programmer Defined Data Types Structured data objects: Structured data objects & data types, specification & implementation of structured data types, Declaration & type checking of data structure, vector & arrays, records Character strings, variable size data structures, Union, pointer & programmer defined data objects, sets, files.

Subprograms and Programmer Defined Data Types: Evolution of data type concept abstraction, encapsulation & information hiding, Subprograms, type definitions, abstract data types, over loaded subprograms, generic subprograms.

Unit–III:

Sequence Control and Data Control Sequence Control: Implicit & explicit sequence control, sequence control within expressions, sequence control within statement, Subprogram sequence control: simple call return, recursive subprograms, Exception & exception handlers, co routines, sequence control.

Data Control: Names & referencing environment, static & dynamic scope, block structure, Local data &local referencing environment, Shared data: dynamic & static scope, Parameter & parameter transmission schemes.

Unit-IV:

Storage Management and Programming Languages Storage Management: Major run time elements requiring storage, programmer and system controlled storage management & phases, Static storage management, Stack based storage management, Heap storage management, variable & fixed size elements.

Programming Languages: Introduction to procedural, non-procedural, structured, logical, functional and object oriented programming language, Comparison of C & C++ programming languages.

Text Books:

1.Terrence W. Pratt,Marvin V. Zelkowitz, Programming Languages design & Implementation, Pearson.
2.Allen Tucker & Robert Noonan, Programming Languages–Principles and Paradigms, Tata McGraw-Hill, 2009.

Reference Books:

1.Ellis Horowitz, Fundamentals of Programming Languages, Galgotia Publications, 2010.
2.C. Ghezzi, Programming Languages Concepts, Wiley Publications, 2010.

Paper Setter's Note: 8 questions of 15 marks each distributed in four sections are to be set taking two from each unit. The candidate is required to attempt five questions in all, taking at least one from each of the four sections.

IT-206 N	Object Oriented Programming Using C++					
Lecture	Tutorial	Practical	Major Test	Minor Test	Total	Time
3	1	-	75	25	100	3 Hour
Purpose	To introduce the principles and paradigms of OOPS for design and implement the Object Oriented System					
Course Outcomes (CO)						
CO 1	To introduce the basic concepts of object oriented programming language and the its representation					
CO 2	To allocate dynamic memory, access private members of class and the behavior of inheritance and its implementation.					
CO 3	To introduce polymorphism, interface design and overloading of operator.					
CO 4	To handle backup system using file, general purpose template and handling of raised exception during programming					

Unit–1:

Introduction to C++, C++ Standard Library, Basics of a Typical C++ Environment, Pre-processors Directives, Illustrative Simple C++ Programs. Header Files and Namespaces, libraryfiles. Concept of objects, basic of object modeling, object classes, associations, behaviors, description, Object Oriented Analysis & Object Modeling techniques,

Object Oriented Concepts : Introduction to Objects and Object Oriented Programming, Encapsulation (Information Hiding), Access Modifiers: Controlling access to a class, method, or variable(public, protected, private, package), Other Modifiers, Polymorphism: Overloading,, Inheritance, Overriding Methods, Abstract Classes, Reusability, Class's Behaviors.

Classes and Data Abstraction: Introduction, Structure Definitions, Accessing Members of Structures, Class Scope and Accessing Class Members, Separating Interface from Implementation ,Controlling Access Function And Utility Functions, Initializing Class Objects: Constructors, Using Default Arguments With Constructors, Using Destructors, Classes : Const(Constant) Object And Const Member Functions, Object as Member of Classes, Friend Function and Friend Classes, Using This Pointer, Dynamic Memory Allocation with New and Delete, Static Class Members, Container Classes And Integrators, Proxy Classes, Function overloading.

Unit-2:

Operator Overloading: Introduction, Fundamentals of Operator Overloading, Restrictions On Operators Overloading, Operator Functions as Class Members vs. as Friend Functions, Overloading, <<, >> Overloading Unary Operators, Overloading Binary Operators.

Inheritance: Introduction, Inheritance: Base Classes And Derived Classes, Protected Members, Casting Base- Class Pointers to Derived- Class Pointers, Using Member Functions, Overriding Base –Class Members in a Derived Class, Public, Protected and Private Inheritance, Using Constructors and Destructors in derived Classes, Implicit Derived –Class Object To Base-Class Object Conversion, Composition Vs. Inheritance.

Unit–3:

Virtual Functions and Polymorphism: Introduction to Virtual Functions, Abstract Base Classes And Concrete Classes, Polymorphism, New Classes And Dynamic Binding, Virtual Destructors, Polymorphism, Dynamic Binding.

Files and I/O Streams: Files and Streams, Creating a Sequential Access File, Reading Data From A Sequential Access File, Updating Sequential Access Files, Random Access Files, Creating A Random Access File, Writing Data Randomly To a Random Access File, Reading Data Sequentially from a Random Access File. Stream Input/Output Classes and Objects, Stream Output, Stream Input, Unformatted I/O (with read and write), Stream Manipulators, Stream Format States, Stream Error States.

Unit-4:

Templates & Exception Handling: Function Templates, Overloading Template Functions, Class Template, Class Templates and Non-Type Parameters, Templates and Inheritance, Templates and Friends, Templates and Static Members.Introduction, Basics of C++ Exception Handling: Try Throw, Catch, Throwing an Exception, Catching an Exception, Re-throwing an Exception, Exception specifications, Processing Unexpected Exceptions, Constructors, Destructors and Exception Handling, Exceptions and Inheritance.

Paper Setter's Note: 8 questions of 15 marks each distributed in four sections are to be set taking two from each unit. The candidate is required to attempt five questions in all, taking at least one from each of the four sections.

Text Books:

- C++ How to Program by H M Deitel and P J Deitel, 1998, Prentice Hall
- Object Oriented Programming in Turbo C++ by Robert Lafore ,1994, The WAITE Group Press.
- Programming with C++ By D Ravichandran, 2003, T.M.H

Reference books:

- Object oriented Programming with C++ by E Balagurusamy, 2001, Tata McGraw-Hill
- Computing Concepts with C++ Essentials by Horstmann, 2003, John Wiley,
- The Complete Reference in C++ By Herbert Schildt, 2002, TMH.

IT-208 N	Operating Systems					
Lecture	Tutorial	Practical	Major Test	Minor Test	Total	Time
3	1	-	75	25	100	3
Purpose	To familiarize how an operating system controls the computer					
Course Outcomes						
CO 1	To study about the process of Operating System and it's scheduling.					
CO 2	To learn about interprocess communication and deadlocks.					
CO 3	To learn about memory management and file system of operating system.					
CO 4	To learn about distributed system and device management.					

UNIT 1:

Introductory Concepts: Operating System functions and characteristics, historical evolution of operating systems, Real time systems, Distributed systems, Methodologies for implementation of O/S service , system calls, system programs , interrupt mechanisms.

Processes: Processes model, process states, process hierarchies, implementation of processes, data structures used such as process table, PCB creation of processes, context switching, exit of processes.

Process scheduling: objective, preemptive Vs non- preemptive scheduling, comparative assessment of different algorithms such as round robin, priority bases scheduling, FCFS, SJF, multiple queues with feedback.

UNIT 2:

Interprocess communication: Race conditions, critical sections, problems of mutual exclusion, Peterson's solution, producer-consumer problem, semaphores, counters, monitors, message passing.

Deadlocks: conditions, modeling, detection and recovery, deadlock avoidance, deadlock prevention.

UNIT 3:

Memory Management: Multiprogramming with fixed partition, variable partitions, virtual partitions, virtual memory, paging, demand paging design and implementation issues in paging such as page tables,
inverted page tables, page replacement algorithms, page fault handling, working set model, local vs global allocation, page size, segmentation and paging.

File Systems: File type, attributes, access and security, file operations, directory structures, path names, directory operations, implementation of file systems, implementation of file and file operations calls, implementation of directories, sharing of files, disk space management, block allocation, free space management, logical file system, physical file system.

UNIT 4:

Device Management: Techniques for device management , dedicated devices, shred devices, virtual devices, device characteristics-hardware considerations: input and output devices, storage devices, independent device operation, buffering, multiple paths, device allocation considerations.

Distributed Systems: Introduction to II/W and S/W concepts in distributed systems, Network operating systems and NFS, NFS architecture and protocol, client- server model, distributed file systems, RPC- Basic operations, parameter passing, RPC semantics in presence of failures threads and thread packages.

Books recommended:

1. Peterson J L &Silberschatz , " Operating System concepts" Addison Wesley
2. Brinch, Hansen, "Operating System Principles" PHI
3. Tenanbaum A S " Operating System", PHI.

Paper Setter's Note: 8 questions of 15 marks each distributed in four sections are to be set taking two from each unit. The candidate is required to attempt five questions in all, taking at least one from each of the four sections.

IT-210 N	Fundamentals of Microprocessor Interfacing & Application					
Lecture	Tutorial	Practical	Major Test	Minor Test	Total	Time
3	1	-	75	25	100	3
Purpose	To learn the architecture and programming of Intel family microprocessors and its interfacing.					
Course Outcomes						
CO 1	To study the Architecture of 8085 microprocessors					
CO 2	Familiarization with the instruction / commands of 8085					
CO 3	Introduction to interfacing of microprocessor					
CO 4	Concept of data transfer among various peripheral devices					

Unit 1

Introduction of Microcomputer System
Introduction to Microcomputer based Systems
Architecture of 8-bit Microprocessor: Intel 8085 microprocessor, Pin description ; Internal architecture , Bus , register organization , Memory organization , Flags , stack , Timing and control unit, instruction cycle , machine cycle , Timing diagram for Fetch and Memory read / write

Unit 2

Programming of 8085
Instruction and data formats ; Instruction Set of 8085 ; introduction to Assembly Language Programming; Stacks and Subroutines ; counter and time delay .

Unit 3

Interfacing I/O devices
Basic interfacing concept ; Interfacing output displays ; Interfacing input devices ; Memory Mapped I/O ; Interrupt structure of 8085

Unit 4

Peripheral devices
An introduction to following devices :- a) Programmable Communication interface (8251) ; b) Programmable Peripheral Interface (8255) ; c) DMA controller (8237) , d) Programmable keyboard / Display interface (8279)
Microprocessor application : Interfacing of LCD , matrix keyboard , stepper motor, Introduction to Microprocessor Controlled Temperature System (MCTS)

Books
1. Ramesh S. Gaonkar, "Microprocessor Architecture, Programming and Application with the 8085", Penram International Publishing (India).
2. B Ram , “Fundamentals of Microprocessors And Microcontrollers” , Dhanpat Rai & sons

Reference Books
1. K. Ray and K M Bhurchandi, “Advanced Microprocessors and Peripherals”, Tata McGraw-Hill
2. K Udaya Kumar , “The 8085 Microprocessor: Architecture, Programming and Interfacing” , Pearson education
3. N.K.Srinath , “8085 Microprocessor: Programming and interfacing” PHI 2005

Paper Setter’s Note: 8 questions of 15 marks each distributed in four sections are to be set taking two from each unit. The candidate is required to attempt five questions in all, taking at least one from each of the four sections.

Programming with C++ Lab
Paper code - IT-212 N

L	T	P
-	-	2

Sessional: 40 Marks
Exam: 60 Marks
Total: 100 Marks
Duration of Exam: 3 Hrs.

LIST OF EXPERIMENTS

Q1. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power ( ) that takes a double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.

Q2. A point on the two two numbers can represent dimensional plane: an X coordinate and a Y coordinate. For example, (4,5) represents a point 4 units to the right of the origin along the X axis and 5 units up the Y axis. The sum of two points can be defined as a new point whose X coordinate is the sum of the X coordinates of the points and whose Y coordinate is the sum of their Y coordinates. Write a program that uses a structure called point to model a point. Define three points, and have the user input values to two of them. Than set the third point equal to the sum of the other two, and display the value of the new point. Interaction with the program might look like this:
Enter coordinates for P1: 3 4
Enter coordinates for P2: 5 7
Coordinates of P1 + P2 are : 8, 11

Q 3. Create the equivalent of a four function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. (It should use a switch statement to select the operation). Finally it should display the result. When it finishes the calculation, the program should ask if the user wants to do another calculation. The response can be 'Y' or 'N'. Some sample interaction with the program might look like this.
Enter first number, operator, second number: 10/ 3
Answer = 3.333333
Do another (Y/ N)? Y
Enter first number, operator, second number 12 + 100
Answer = 112
Do another (Y/ N) ? N

Q4. A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767) and the number (8900). Write a program that
uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user
input a number for the other one. Then display both numbers. The interchange might look like this:
Enter your area code, exchange, and number: 415 555 1212
My number is (212) 767-8900
Your number is (415) 555-1212

Q 5. Create two classes DM and DB which store the value of distances. DM stores distances in metres and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or metres and cenitmetres depending on the object on display.

Q 6. Create a class rational which represents a numerical value by two double values- NUMERATOR & DENOMINATOR. Include the following public member Functions:
• constructor with no arguments (default).
• constructor with two arguments.
• void reduce( ) that reduces the rational number by eliminating the highest common factor between the numerator and denominator.
• Overload + operator to add two rational number.
• Overload >> operator to enable input through cin.

• Overload << operator to enable output through cout.
Write a main ( ) to test all the functions in the class.

Q 7. Consider the following class definition

```
class father {
protected :int age;
public;
father (int x) {age = x;}
virtual void iam ( )
{ cout<< "I AM THE FATHER, my age is : "<< age<< end1:}
};
```

Derive the two classes son and daughter from the above class and for each, define I am ( ) to write our similar but appropriate messages. You should also define suitable constructors for these classes. Now, write a main ( ) that creates objects of the three classes and then calls I am ( ) for them. Declare pointer to father. Successively, assign addresses of objects of the two derived classes to this pointer and in each case, call I am ( ) through the pointer to demonstrate polymorphism inaction.

Q 8. Write a program that creates a binary file by reading the data for the students from the terminal.
`The data of each student consist of roll no., name (a string of 30 or lesser no. of characters) and marks.

Q9. A hospital wants to create a database regarding its indoor patients. The information to store include
a) Name of the patient
b) Date of admission
c) Disease
d) Date of discharge
Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to
enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about all the to store the
age of the patients. List the information about all the pediatric patients (less than twelve years in age).

Q 10. Make a class **Employee** with a name and salary. Make a class **Manager** inherit from **Employee**. Add an instance variable, named department, of type string. Supply a method to **to String** that prints the manager's name, department and salary. Make a class **Executive** inherit from **Manager**. Supply a method **to String** that prints the string **"Executive"** followed by the information stored in the **Manager** superclass object. Supply a test program that tests these classes and methods.

Q11. Imagine a tollbooth with a class called toll Booth. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to 0. A member function called paying Car ( ) increments the car total and adds 0.50 to the cash total. Another function, called nopayCar ( ), increments the car total but adds nothing to the cash total. Finally, a member function called displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the ESC kay should cause the program to print out the total cars and total cash and then exit.

Q12. Write a function called reversit ( ) that reverses a string (an array of char). Use a for loop that swaps the first and last characters, then the second and next to last characters and so on. The string should be passed to reversit ( ) as an argument. Write a program to exercise reversit ( ). The program should get a string from the user, call reversit ( ), and print out the result. Use an input method that allows embedded blanks. Test the program with Napoleon's famous phrase, "Able was I ere I saw Elba)".

Q13. Create some objects of the string class, and put them in a Deque-some at the head of the Deque and some at the tail. Display the contents of the Deque using the for Each( ) function and a user written display function. Then search the Deque for a particular string, using the first That ( ) function and display any strings that match. Finally remove all the items from the Deque using the getLeft ( ) function and display each item. Notice the order in which the items are displayed: Using getLeft ( ), those inserted on the left (head) of the Deque are removed in "last in first out" order while those put on the right side are removed in "first in first out" order. The opposite would be true ifgetRight ( ) were used.

Q 14. Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class account that stores customer name, account number and type of account. From this derive the classescur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks:

a) Accept deposit from a customer and update the balance.
b) Display the balance.
c) Compute and deposit interest.
d) Permit withdrawal and update the balance.
e) Check for the minimum balance, impose penalty, necessary and update the balance.
f) Do not use any constructors. Use member functions to initialize the class members.

Q 15. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data ( ) to initialize baseclass data members and another member function display_area ( ) to compute and display the area of figures. Make display_area ( ) as a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangles and used as follows:

Area of rectangle = x * y
Area of triangle = ½ * x * y

.....

Microprocessors and Interfacing Lab
Paper code- IT-214 N

L T P
- - 2

Sessional: 40 Marks
Exam: 60 Marks
Total: 100 Marks
Duration of Exam: 3 Hrs.

LIST OF EXPERIMENTS

1. Study of 8085 Microprocessor kit.
2. Write a program using 8085 and verify for :
   a. addition of two 8-bit numbers result is 8 bit
   b. addition of two 8-bit numbers result is 16 bit.
3. Write a program using 8085 and verify for :
   a. 8-bit subtraction
   b. 16-bit subtraction
4. Write a program using 8085 for multiplication of two 8- bit numbers by repeated addition method. Check for minimum number of additions and test for typical data.
5. Write a program using 8085 for multiplication of two 8- bit numbers by bit rotation method
6. Write a program using 8085 for division of two 8- bit
7. Write a program using 8085 for dividing two 8- bit numbers by bit rotation method and test for typical data.
8. Shift an 8 bit number left by 2 bits.
9. Find 2's compliment of an 8bit and 16 bit number
10. To find larger of two numbers.
11. To find square-root of a number
12. Rolling display "HELLO WORLD" on the address and data field of screen
13. Write a program to control the operation of stepper motor using 8085
14. Write a program to interface adc & dac with 8085 & demonstrate generation of square wave.

Note : A student has to perform at least ten experiments. Seven experiments should be performed from the above list. Remaining three experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus.

Computer Hardware & Troubleshooting Lab
Paper code - IT-216 N

L	T	P
-	-	2

Sessional: 40 Marks
Exam: 60 Marks
Total: 100 Marks
Duration of Exam: 3 Hrs.

LIST OF EXPERIMENTS

1. To solder and de-solder various components.
2. To check and measure various supply voltages of PC.
3. To make comparative study of motherboards: 386, 486, PI, PII, PIII.
4. To observe and study various various cables, connections and parts used in computer communication.
5. To study various cards used in a system viz., display card, LAN card etc.
6. To remove, study and replace floppy disk drive.
7. To remove, study and replace hard disk.
8. To remove, study and replace CD ROM drive.
9. To study monitor, its circuitry and various presets and some elementary fault detection.
10. To study printer assembly and elementary fault detection of DMP and laser printers.
11. To observe various cables and connectors used in networking.
12. To study parts of keyboard and mouse.
13. To assemble a PC.
14. Troubleshooting exercises related to various components of computer like monitor,drives, memory, and printers etc.

Books

1. Mark Mines Complete PC upgrade & maintenance guide. BPB publications.
2. Craig Zacker & John Rouske, PC Hard ware : The Complete Reference TMH.
3. Scott Mueller, Upgrading and Repairing PCs. PHI, 19

Programming with MATLAB
Paper code - IT-218 N

L	T	P
-	-	2

Sessional: 40 Marks
Exam: 60 Marks
Total: 100 Marks
Duration of Exam: 3 Hrs.

1. To study MATLAB environment and to familiarize with Command Window, History, Workspace, Current Directory, Figure window, Edit window, Shortcuts, Help files.
2. Data types, Constants and Variables, Character constants, operators, Assignment statements.
3. Control Structures: For loops, While, If control structures, Switch, Break, Continue statements Input-Output functions, Reading and Storing Data.
4. Write a MATLAB program to calculate the following expression and round the answers to the nearest integer.
   a) $z = 5x^2 + y^2$ where $x = 2$, $y = 4$
   b) $z = 3\sin(x) + 4\cos(x) + 3e^y$ where $x = \pi/3$, $y = 2$
5. Vectors and Matrices, commands to operate on vectors and matrices, matrix Manipulations.
6. Arithmetic operations on Matrices, Relational operations on Matrices, Logical operations on Matrices.
7. If x = [1 4; 8 3], find :
   a) the inverse matrix of x .
   b) the transpose of x.
   c) Determinant of x
8. Polynomial Evaluation, Roots of Polynomial, Arithmetic operations on Polynomials.
9. Graphics: 2D plots, Printing labels, Grid & Axes box, Text in plot, Bar and Pie chart
10. To plot a sine wave of frequency 1Khz
11. Study of Simulink
12. To implement a simple calculator as a GUI
13. Solve the following system
    $x + y - 2z = 3$
    $2x + y = 7$
    $x + y - z = 4$
14. Write a program to read three bits x, y, z, then compute:
    a) v = (x and y) or z
    b) w = not (x or y) and z
15. Represent the following complex numbers in polar coordinate
    Z= 3+ 4j

Note: A student has to perform at least ten experiments. Seven experiments should be performed from the above list. Remaining experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus.

MPC- 201N ENVIRONMENTAL STUDIES

L	T	P	Sessional	Exam	Time
3	-	-	25	75	3H

UNIT I

The multidisciplinary nature of environmental studies. Definition, Scope and Importance. Need for public awareness. Natural Resources: Renewable and Non-Renewable Resources: Natural resources and associated problems.

(a) Forest Resources: Use and over-exploitation, deforestation, case studies. Timber extraction, mining, dams and their effects on forests and tribal people.

(b) Water Resources- Use and over-utilization of surface and ground water, floods, drought, conflicts over water, dams-benefits and problems.

(c) Mineral Resources- Use and exploitation, environmental effects of extracting and using mineral resources, case studies.

(d) Food Resources- World Food Problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies.

(e) Energy Resources- Growing energy needs, renewable and non-renewable energy sources, use of alternate energy sources, case studies.

(f) Land Resources- Land as a resource, land, degradation, man induced landslides, soil erosion and desertification.

Role of an individual in conservation of natural resources. Equitable use of resources for sustainable lifestyle.

UNIT II

Ecosystem- Concept of an ecosystem, Structure and function of an ecosystem, Producers, consumers and decomposers, Energy flow in the ecosystem. Ecological succession, Food Chains, food webs and ecological pyramids. Introduction, types, characteristic features, structure and function of the following ecosystem.

a. Forest Ecosystem
b. Grassland Ecosystem
c. Desert Ecosystem
d. Aquatic Ecosystems (ponds, streams, lakes, rivers, oceans, esturaries

Field Work: Visit to a local area to document Environment assets-river/forest/grassland/ hill/ mountain. Visit to a local polluted site- Urban /Rural/Industrial/Agricultural. Study of common plants, insects and birds. Study of simple ecosystems-pond, river, hill, slopes etc. (Field work equal to 5 lecture hours).

UNIT III

Biodiversity and its conservation. Introduction, Definition: genetic, species and ecosystem diversity. Biogeographical classification of India. Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values. Biodiversity of global, National and local levels. India as a mega-diversity nation Hot spots of Biodiversity. Threats to biodiversity: Habitat loss, poaching of wild life, man-wildlife conflicts. Endangered and endemic species of India. Conservation of Biodiversity- In situ and Ex-Situ conservation of biodiversity.

Environmental Pollution: Definition, Cause, effects and control measures of- (a) Air Pollution (b) Water Pollution (c) Soil Pollution (d) Marine Pollution (e) Noise Pollution (f) Thermal Pollution (g) Nuclear Hazards

Solid waste management- cause, effects and control measures of urban and industrial wastes. Role of an individual in prevention of pollution. Pollution case studies. Disaster management: floods, earthquake, cyclone and landslides

UNIT IV

Social Issues and the Environment, From unsustainable to sustainable development, Urban problems related to energy, Water conservation, rain water harvesting, watershed management.

Resettlement and rehabilitation of people: Its problems and concerns. Case Studies. Environmental ethics-issues and possible solutions, Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust, Case studies.

Wasteland Reclamation, Consumerism and waste products, Environment Protection Act, Air (Prevention and Control of Pollution) Act, Water (Prevention and Control of Pollution) Act, Wildlife Protection Act, Forest Conservation Act, Issues involved in enforcement of environmental legislation, Public Awareness, Human population and the Environment, Population growth, variation among nations. Population explosion-Family Welfare Programme, Environment and human health, Human rights, Value Education, HIV/AIDS, Women and Child Welfare, Role of Information Technology in Environment and Human Health, Case Studies.

Suggested Text Books & References:

1. Environmental Studies- Deswal and Deswal. Dhanpat Rai & Co.
2. Environmental Science & Engineering Anandan, P. and Kumaravelan, R. 2009. Scitech Publications (India) Pvt. Ltd., India
3. Environmental Studies. Daniels Ranjit R. J. and Krishnaswamy. 2013. Wiley India.
4. Environmental Science- Botkin and Keller. 2012. Wiley, India.