UNIVERSITY OF PUNE

For Computer Engineering & Information Technology (Sem II)

207003 ENGINEERING MATHEMATICS – III (2012 Course)

Teaching Scheme:
Lectures – 4 Hrs./Week
Tutorials – 1 Hr./Week

Examination Scheme:
Paper – 50 Marks (2 Hrs.)
Online – 50 Marks
Term work: 25 Marks

Section I

Unit I: Linear Differential Equations (**LDE**) and Applications (09 Hours)

LDE of n^{th} order with constant coefficients, Method of variation of parameters, Cauchy's & Legendre's DE, Simultaneous & Symmetric simultaneous DE. Modeling of Electrical circuits.

Unit II: Transforms (09 Hours)

Fourier Transform (**FT**): Complex Exponential form of Fourier series, Fourier integral theorem, Sine & Cosine integrals, Fourier Transform, Fourier Sine and Cosine Transform and their inverses.

Z - Transform (**ZT**): Introduction, Definition, Standard properties, ZT of standard sequences and their inverses. Solution of difference equations.

Unit III: Statistics and Probability (09 Hours)

Measures of central tendency, Standard deviation, Coefficient of variation, Moments, Skewness and Kurtosis, Correlation and Regression, Reliability of Regression Estimates.

Probability, Probability density function, Probability distributions: Binomial, Poisson, Normal and Hypergometric. Test of Hypothesis: Chi-Square test.

Section II

Unit IV: Vector Differential Calculus (09 Hours)

Physical interpretation of Vector differentiation, Vector differential operator, Gradient, Divergence and Curl, Directional derivative, Solenoidal, Irrotational and Conservative fields, Scalar potential, Vector identities.

Unit V: Vector Integral Calculus and Applications (09 Hours)

Line, Surface and Volume integrals, Work-done, Green's Lemma, Gauss's Divergence theorem, Stoke's theorem. Applications to problems in Electro-magnetic fields.

Unit VI: Complex Variables (09 Hours)

Functions of Complex variables, Analytic functions, Cauchy-Riemann equations, Conformal mapping, Bilinear transformation, Cauchy's integral theorem, Cauchy's integral formula, Laurent's series, Residue theorem.

Text Books:

1. Advanced Engineering Mathematics, 9e, by Erwin Kreyszig (Wiley India).
2. Advanced Engineering Mathematics, 7e, by Peter V. O'Neil (Cengage Learning).

Reference Books:

1. Advanced Engineering Mathematics, 2e, by M. D. Greenberg (Pearson Education).
2. Advanced Engineering Mathematics, Wylie C.R. & Barrett L.C. (McGraw-Hill, Inc.)
3. Higher Engineering Mathematics by B. S. Grewal (Khanna Publication, Delhi).
4. Applied Mathematics (Volumes I and II) by P. N. Wartikar & J. N. Wartikar (Pune Vidyarthi Griha Prakashan, Pune).
5. Higher Engineering Mathematics by B.V. Ramana (Tata McGraw-Hill).
6. Advanced Engineering Mathematics with MATLAB, 2e, by Thomas L. Harman, James Dabney and Norman Richert (Brooks/Cole, Thomson Learning).

Tutorial and Term Work:

i) Tutorial for the subject shall be engaged in minimum of four batches (batch size of 20 students maximum) per division.

ii) Term work shall consist of six assignments (one per each unit) based on performance and continuous internal assessment.

210247 OBJECT ORIENTED AND MULTICORE PROGRAMMING

Teaching Scheme
Lectures: 3 Hrs/week
Practical: 2Hrs/Week

Examination Scheme
Theory: 50 Marks
On-Line: 50 Marks
Practical: 50 Marks
Term Work: 25 Marks

Prerequisite: Data Structures and problem Solving (Subject Code: 210244)

Course Objectives

1. To study the representation, implementation and applications of data structures
2. To study implementation of data structures using OOP concepts
3. To compare the benefits of static and dynamic data structures
4. To choose the appropriate data structure for modeling a given problem

Course Outcomes

1. Ability to use overloading and morphism in programming
2. Ability to understand basics of multicore-programming
3. Ability to perform programming using threads

Unit I	**Foundations of Object Oriented Programming**	**(9 Hrs)**
	Introduction to procedural, modular, object-oriented and generic programming techniques, Limitations of procedural programming, Need of object-oriented programming, fundamentals of object-oriented programming: objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism,	3 Hrs
	++: Extensions to C: Variable declarations, global scope, 'const', reference variables, comments, default parameters, function prototypes, function overloading, inline functions, default and constant arguments, 'cin', 'cout', formatting and I/O manipulators, new and delete operators	2 Hrs
	Defining a class, data members and methods, public, private and protected members, inline member functions, static data members, static member functions, 'this' pointer, constructors, destructors, friend function, dynamic memory allocation, array of objects, pointers and classes, class as ADTs and code reuse	4 Hrs
Unit II	**Overloading and Inheritance**	**(7 Hrs)**
	Introduction, Need of operator overloading, overloading the assignment, binary and unary operators, overloading using friends, rules for operator overloading, type conversions	(3Hrs)
	Concept and need, single inheritance, base and derived classes, friend classes, types of inheritance, hybrid inheritance, member access control, static class, multiple inheritance, ambiguity, virtual base class, polymorphism, virtual functions, pure virtual functions, abstract base class, virtual destructors, early and late binding, container classes	(4 Hrs)
Unit III	**Templates and Exception Handling**	**(7 Hrs)**
	Introduction, Templates: Function template and class template, function overloading vs. function templates, member function templates and template arguments, Introduction to Generic Programming: Introduction to Standard Template Library (STL), containers, iterators and algorithms, study of container template classes for vectors and stacks and related algorithms, **Namespaces:** Introduction, Rules of namespaces	(5 Hrs)
	Exception Handling: Introduction, syntax for exception handling code: try-catch-	(2Hrs)

	throw, Multiple Exceptions, Exceptions with arguments, Introduction to RTTI **Managing Console I/O Operations:** Introduction, C++ streams, stream classes, unformatted I/O, formatted I/O and I/O manipulators	
Unit IV	**Challenges in Multi-core Programming**	**(4 Hrs)**
	Sequential Models, Concurrency, Challenges for software development, New libraries for C++ developers, Processor Architecture Challenges, Operating systems (OS) roll in concurrent development: Consistent interfaces, Resource Management, OS interaction, Core OS services, Application Program Interfaces, Decomposition and Operating systems Roll, Hiding the Operating systems Roll: Abstraction and Encapsulation, Interface classes for POSIX API	
	Processes, Interface classes and predicates	**(8 Hrs)**
	Multicore and multiprocessors, Processes and threads, Parent-child relations, Process control block, Anatomy of a process, Process States, Process Scheduling, ps utility, process priorities, Context switch, Activities in process creation, Process environment variables, using system() to Spawn Processes, Killing a process, Process Resources, Asynchronous and Synchronous Processes, the wait() function call, Predicates, Processes and Interface classes	
Unit V	**Multithreading**	**(7 Hrs)**
	Thread, User and Kernel level threads, Thread Context, Hardware Threads and software threads, Comparing threads to processes, setting thread attributes, The architecture of a Thread, Compiling and linking threaded programs, Creating threads, Managing threads, Thread Interface classes	
Unit VI	**Communication and synchronization of Concurrent tasks**	**(6 Hrs)**
	Communication and synchronization, Synchronizing concurrency, Thread Strategy approaches, Decomposition and Encapsulation of Work Case studies of concurrency models	

Text Books(TB):

1. R. Gilberg, B. Forouzan, “Data Structures: A pseudo code approach with C”, Cengage Learning, ISBN 9788131503140.
2. A. Michael Berman, “Data structures via C++”, Oxford University Press, 2002, ISBN-0-19-510843-4.
3. Professional Multicore Programming: Design and implementation for C++ Developers, Wiley India Edition, ISBN: 978-81-265-1875-3,

Reference Books(RB):

1. E. Horowitz, S. Sahni, D. Mehta “Fundamentals of Data Structures in C++”, Galgotia Book Source, New Delhi, 1995, ISBN 16782928.
2. Y. Langsam, M. Augenstin and A. Tannenbaum, “Data Structures using C and C++”, 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.
3. R. Gilberg, B. Forouzan, “Data Structures: A pseudo code approach with C++”, Cengage Learning, ISBN 9788131504925.
4. A. Tharp ,”File organisation and processing”,2008 ,Willey India edition ,9788126518685
5. A. Drozdek, “Data Structures in C++”, 2nd Edition, Thomson Brookes /COLE Books, 2002, ISBN 981 – 240 – 079 – 6.
6. J. Tremblay, P. Soresan, “An introduction to data structures with Applications”, 2nd edition, Tata McGraw-Hill International Editions, 1984, ISBN-0-07-462471-7.
7. M. Folk, B. Zoellick, G. Riccardi, “File Structure An Object oriented approach with C++”, Pearson Education, 2002, ISBN 81 – 7808 – 131 – 8.
8. M. Weiss, “Data Structures and Algorithm Analysis in C++”, 2nd edition, Pearson Education, 2002, ISBN-81-7808-670-0

Laboratory Assignments

List of Practical Assignments:

Tools		
1.	Operating Systems (64-Bit)	Latest 64-BIT Version and update of Microsoft Windows 7/ Windows 8 Operating System onwards or 64-bit Open source Linux or its derivative
2.	Programming Tools (64-Bit)	Any 64 bit C++ Programming tool like TC++/VC++/BC++/G++/GCC, JAVA, Python
3.	Assignment Allocation	All assignments from groups A,B,C given below must be covered in a batch of 15 students or AICTE specified Student : Teacher ratio. Submission journal will have assignments from group A, atleast 3 assignments from group B and atleast 1 assignment from group C covering all assignments of the groups A, B, C per batch.
4.	Write up	Aim, Index terms, use of discrete mathematics to re-write/describe the problem definition. Use of set theory, Probability Theory or other relevant theory, Discrete Structures to give the problem solution, Data-independence / Dataflow Architecture for the development of turing machine/state Diagram, multiplexer logic to identify opportunity of morphism and overloading and data flow architecture and optimal/effective use of multicore of the CPU. LATEX Soft Copy CD and Handwritten Hard Copy Journal to be submitted as Term-work.

Laboratory Experiments

<table>
<tr><td colspan="2">Group A (Mandatory Assignments)
(Use of Constructor, destructor, Static member functions, friend class, this pointer, inline code and dynamic memory allocation is expected)</td></tr>
<tr><td>1</td><td>1. Create a class named weather report that holds a daily weather report with data members day_of_month,hightemp,lowtemp,amount_rain and amount_snow. The constructor initializes the fields with default values: 99 for day_of_month, 999 for hightemp,-999 for low emp and 0 for amount_rain and amount_snow. Include a function that prompts the user and sets values for each field so that you can override the default values. Write a C++/Java/Python program that creates a monthly report.
d) Menu driven program with options to Enter data and Display report
e) Report Format
<table><tr><td>Day</td><td>Amt_Rain</td><td>Amt_snow</td><td>High_temp</td><td>Low_temp</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Avg</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table></td></tr>
<tr><td>2</td><td>A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise the message “Required copies not in stock” is displayed.</td></tr>
</table>

	Design a system using a class called **books** with suitable member functions and Constructors. Use **new** operator in constructors to allocate memory space required. Implement C++ program for the system.
3	Develop an object oriented program in C++ to create a database of the personnel information system containing the following information: Name, Date of Birth, Blood group, Height, Weight, Insurance Policy, number, Contact address, telephone number, driving license no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, copy, constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete.
4	Design a C++ Class 'Complex ' with data members for real and imaginary part. Provide default and parameterized constructors. Write a program to perform arithmetic operations of two complex numbers using operator overloading (using either member functions or friend functions).
5	Write a C++ program to perform String operations i. = Equality ii. == String Copy iii. + Concatenation iv. << To display a string v. >> To reverse a string vi. Function to determine whether a string is a palindrome To find occurrence of a sub-string. Use Operator Overloading
6	Develop an object oriented program in C++ to create a database of the personnel information system containing the following information: Name, Date of Birth, Blood group, Height, Weight, Insurance Policy number, Contact address, telephone number, driving licence no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, copy constructor, destructor, static member functions , friend class, this pointer, inline code and dynamic memory allocation operators-new and delete.
7	Write a program in C++ using function template to read two matrices of different data types such as integers and floating point values and perform simple arithmetic operations on these matrices separately and display it.
8	Design a C++ base class consisting of the data members such as name of the student, roll number and subject. The derived class consists of the data members subject code, internal assessment and university examination marks. Construct a virtual base class for the item name of the student and roll number. The program should have the facilities. i) Build a master table ii) List a table iii) Insert a new entry iv) Delete old entry v) Edit an entry vi) Search for a record
9	Create a C++ class named Television that has data members to hold the model number and the screen size in inches, and the price. Member functions include overloaded insertion and extraction operators. If more than four digits are entered for the model, if the screen size is smaller than 12 or greater than 70 inches, or if the price is negative or over $5000 then throw an integer. Write a main() function that instantiates a television object, allows user to enter data and displays the data members .If an exception is caught, replace all the data member values with zero values.
Group B (any Two)	
1	A 'C' program function having one IF-THEN-ELSE returns the truth-ness value (TRUE/FALSE) is to be replaced by overloading while porting it to C++. Use appropriate overloading to replace IF-THEN-ELSE. Demonstrate the functioning by

	using it in a class.
2	A ‘C’ program uses a structure to implement a circular linked list for maintaining the numbers in ascending order. New arrival of number increases the size of circular linked list. This program is to be ported to C++ using appropriate C++ Data structures and programming. (In C++ avoid use of structure and IF-Then-Else or while/do-while etc.)
3	Implement C++/Java/Python program to 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 function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make classes to suit their requirements. Using these three classes, design a program that will accept dimension 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 =1/2*x*y
4	Implement C++/Java/Python program to implement a base class consisting of the data members such as name of the student, roll number and subject. The derived class consists of the data members subject code ,internal assessment and university examination marks. The program should have the facilities. i) Build a master table ii) List a table iii) Insert a new entry iv) Delete old entry v) Edit an entry vi) Search for a record. Use virtual functions.
5	Implement C++/Java/Python program to write a class template to represent a generic vector. Include following member functions: To create the vector. To modify the value of a given element To multiply by a scalar value To display the vector in the form (10,20,30,…)
6	Implement C++/Java/Python program for bubble sort using function template
7	Refer the standard template library to use list container and using C++/Java implement following member functions of list class: empty, insert, merge, reverse, sort
8	Write a C++/Java program for the following: 1) A function to read two double type numbers from keyboard 2) A function to calculate the division of these two numbers 3) A try block to throw an exception when a wrong type of data is keyed in 4) A try block to detect and throw an exception if the condition “divide-by-zero” occurs 5) Appropriate catch block to handle the exceptions thrown
9	Write a C++/Java program for the following: Create a class named Television that has data members to hold the model number and the screen size in inches, and the price. Member functions include overloaded insertion and extraction operators. If more than four digits are entered for the model, if the screen size is smaller than 12 or greater than 70 inches, or if the price is negative or over $5000 then throw an exception. Write a main() function that instantiates a television object, allows user to enter data and displays the data members .If an exception is caught ,replace all the data member values with zero

	values.
10	Create employee bio-data using following classes i) Personal record ii))Professional record iii) Academic record Assume appropriate data members and member function to accept required data & print bio-data. Create bio-data using multiple inheritance using C++/Java/Python.
11	Using multi-core programming implement POSIX-spawn() function to create a process
12	Implement a simple interface class for a POSIX Process using multi-core programming
13	Using multi-core programming implement a predicate class
14	Implement POSIX queue class that encapsulates the basic function such as open, send, receive, remove, close. Use multi-core programming
15	Implement POSIX semaphore using multi-core programming
16	Using multi-core programming, implement Mutex semaphore for : Initialization Request ownership Release ownership Try ownership Destruction
17	Using multi-core programming implement a thread interface class
18	Write a Object Oriented Program using C++/Java for 6. passing command line arguments to the thread function 7. using the command line argument to determine the number of threads Use multi-core programming
Group C (Advanced Assignments) Any One	
1.	**Write a concurrent program to implement the Odd-Even Merge Sort. Effective use of Multicore Architecture Core 1 and Core 2 effectively is expected.**
2.	**Write a concurrent program to implement the Dining philosophers problem. Effective use of Multicore Architecture is expected.**
3.	**Write a concurrent program for Matrix Multiplication. Effective use of Multicore Architecture is expected.**

Note: Examination will be based on the assignments performed.

210248 Microprocessor and Interfacing Techniques

Teaching Scheme
Lectures: 3 Hrs/week
Practical: 2Hrs/Week

Examination Scheme
Theory: 50 Marks
OnLine: 50 Marks
Term Work: 25 Marks

Prerequisites: Microprocessor Architecture

Course Objectives:

1. To learn the Peripheral architecture and programming 0f Microprocessor.
2. To learn peripherals and their interfacing with 8086 Microprocessor.
3. To study the DOS Internals.
4. To Study NDP and Design of Microprocessor based System.

Course Outcomes:

1.Ability to handle, interface and program using legacy peripherals
2.Ability to understand I/O Hub functions

Unit I	**Introduction to Interfacing**	
	Introduction to 64/32 bit microprocessors, Differences in Architecture and Pin diagram of 8086, 80386DX, i7 microprocessor, Differences in Programmers model (Register SET) and Segmentation of 8086, 80386DX and i7 microprocessor, logical to physical address translation, BUS Sizing, Memory Read/write (8/16/32/64-bit) cycle timing diagrams, I/O Read /Write cycles timing diagrams, Address mapping and decoding. Addressing modes, Instruction set of 80386 in detail, Instruction Formats, Stacks, Assembly Language Programming, Assembler, Linker, Debugger (Turbo debugger), Directives, Procedures (Near & Far), Macros, Loop constructs, 80386 Programming examples.	**(8 Hrs)**
Unit II	**Legacy DOS Internals and Interrupt Controller**	
	Interrupt Structure, Interrupt Vector Table (IVT), ISR, Hardware and software Interrupts Internals of DOS, DOS loading, DOS memory map, Internal and external commands of DOS, BIOS & DOS Interrupts. Concepts of PSP, .EXE & .COM files, Concepts of TSR, **8259** (Programmable Interrupt Controller): Features, Block Diagram, Control & status registers, Interfacing & Programming. Computer Memory and Memory Mapping in 64-bit mode	**(8 Hrs)**
Unit III	**Legacy Serial and parallel Communication**	
	(Programmable Peripheral Interface), Serial Communication- Synchronous & Asynchronous, **8251**(USART): Features, Block Diagram, Control & status registers, Operating modes, Interfacing & Programming (8255 and 8251) Concept of ADC -Successive Approximation & Interfacing, Concept of DAC R-2R (ladder) & Interfacing, Introduction to Sensors & Transducers, Keyboard Display & Centronics Printer Parallel Interfacing, USB Printers	**(9 Hrs)**
Unit IV	**Legacy Timer and DMA Controller**	
	Keyboard and Display Controller, **8253** (Programmable Interval Timer): Features, Block Diagram, Control & status registers, Operating modes, Interfacing & Programming. Concept of DMA, **8237** DMA Controller: Features, Block Diagram	**(7 Hrs)**
Unit V	**Legacy Math-Controller**	
	Minimum & Maximum mode of 8086, Support chips 8282,8284,8286,8288 **8087(NDP) -** Features, Block Diagram, Control & status registers, typical Instruction Set & Programming Detail Design of 8086 based minimum system with EPROM,	**(8 Hrs)**

	SRAM & Peripherals such as 8255,8253,8251,8279 with keyboard & seven segments Display.	
Unit VI	**Advanced I/O Controllers**	
	Introduction to Intel 58X chipset, Intel® 82801IJR I/O Controller Hub, Study of Intel i5 Motherboard Block Diagram	**(8 Hrs)**

Text Books:

1. Douglas Hall, "Microprocessors & Interfacing", McGraw Hill, Revised 2nd Edition, 2006 ISBN 0-07-100462-9
2. John Uffenbeck," The 8086/88 Family: Design, Programming & Interfacing", PHI,
3. A.Ray, K.Bhurchandi, "Advanced Microprocessors and peripherals: Arch, Programming & Interfacing", Tata McGraw Hill,2004 ISBN 0-07-463841-6
4. Introduction to 64 bit Intel Assembly Language Programming for Linux, 2nd Edition, Ray Seyfarth, ISBN10: 1478119209, ISBN-13: 9781478119203, 2012

References Books:

1. Liu, Gibson, "Microcomputer Systems: The 8086/88 Family", 2nd Edition, PHI,2005
2. Kenneth Ayala, "The 8086 Microprocessor: Programming & Interfacing the PC", Cengage Learning, Indian Edition, 2008
Ray Dunkon, "Advanced MSDOS Programming", 2nd Edition, BPB Publication.
3. Kip Irvine, "Assembly language for IBM PC", PHI, 2nd Edition, 1993
4. Peter Abel, "Assembly language programming", Pearson Edu,5th Edition,2002
5. Intel Microprocessor and peripheral Handbook: Volume 1
6. Yashwant Kanitkar, "TSR through C", BPB Publication, 1995, ISBN 81- 7029-520-3.

MICROPROCESSOR INTERFACING LABORATORY

Suggested List of Assignments

Group A

1. Write X86/64 Assembly language program (ALP) to add array of N hexadecimal numbers stored in the memory. Accept input from the user.
2. Write X86/64 ALP to perform non-overlapped and overlapped block transfer (with and without string specific instructions). Block containing data can be defined in the data segment.
3. Write 64 bit ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for:
(a) HEX to BCD b) BCD to HEX (c) EXIT.
Display proper strings to prompt the user while accepting the input and displaying the result. (use of 64-bit registers is expected)
4. Write X86/64 ALP for the following operations on the string entered by the user. (use of 64-bit registers is expected)
a) Calculate Length of the string b) Reverse the string
c) Check whether the string is palindrome
OR
Make your program user friendly by providing MENU like:
(a) Enter the string b) Calculate length of string c) Reverse string d) Check palindrome e) Exit
Display appropriate messages to prompt the user while accepting the input and displaying the result.
5. Write 8086 ALP to perform string manipulation. The strings to be accepted from the user is to be stored in data segment of program_l and write FAR PROCEDURES in code segment program_2 for following operations on the string:

(a) Concatenation of two strings (b) Number of occurrences of a sub-string in the given string Use PUBLIC and EXTERN directive. Create .OBJ files of both the modules and link them to create an EXE file.
6. Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use successive addition and add and shift method. Accept input from the user. (use of 64-bit registers is expected)
7. Write 8087ALP to obtain:
i) Mean ii) Variance iii) Standard Deviation
For a given set of data elements defined in data segment. Also display result.

Group B

1. 8255

(a) Write 8086 ALP to convert an analog signal in the range of 0V to 5V to its corresponding digital signal using successive approximation ADC and dual slope ADC. Find resolution used in both the ADC's and compare the results.
(b) Write 8086 ALP to interface DAC and generate following waveforms on oscilloscope,
(i) Square wave - Variable Duty Cycle and Frequency.
(ii) Ramp wave - Variable direction, (iii) Trapezoidal wave (iv) Stair case wave
(c) Write 8086 ALP to rotate a stepper motor for given number of steps at a given angle and in the given direction of rotation based on the user choice such as
(i) If 'C' key is pressed - clockwise rotation, (ii) If 'A' key is pressed -
anticlockwise rotation. (iii) If 'B' is pressed - 1/2 clockwise and Vz
Anti-clock wise rotation, (iv) If 'S' key is pressed - stop rotation. Also write routines to accelerate and de-accelerate the motor.
(d)Write 8086 ALP to print a text message on printer using Centronixs parallel printer interface.
NOTE: Select any two from 8255 assignments

2. 8253

Write 8086 ALP to program 8253 in Mode 0, modify the program for hardware retriggerable Mono shot mode. Generate a square wave with a pulse of 1 ms. Comment on the difference between Hardware Triggered and software triggered strobe mode. Observe the waveform at GATE & out pin of 1C 8254 on CRO

3. 8279

Write 8086 ALP to initialize 8279 and to display characters in right entry mode.
Provide also the facility to display
• Character in left entry mode.
• Rolling display.
• Flashing display

4. 8251

Perform an experiment to establish communication between two 8251 systems A and B. Program 8251 system A in asynchronous transmitter mode and 8251 system B in asynchronous receiver mode. Write an ALP to transmit the data from system A and receive the data at system B. The requirements are as follows:

Transmission:

• message is stored as ASCII characters in the memory.
• message specifies the number of characters to be transmitted as the first byte.

Reception:

• Message is retrieved and stored in the memory.
• Successful reception should be indicated.

5. 8259

Write 8086 APL to interface 8259 in cascade mode (M/S) and demonstrate execution of ISR in following manner:
Main program will display two digits up counter. When slave IRQ interrupt occurs, it clears the counter and starts up counting again. When Master IR1 interrupt occurs, it resets the counter to FFH and starts down counting.

6. TSR Program

Write a TSR program in 8086 ALP to implement Real Time Clock (RTC). Read the Real Time from CMOS chip by suitable INT and FUNCTION and display the RTC at the bottom right corner on the screen. Access the video RAM directly in your routine.

7. TSR Program

Write a TSR program in 8086 ALP to implement Screen Saver. Screen Saver should get activated if the keyboard is idle for 7 seconds. Access the video RAM directly in your routine.

Group C

1. Study of Intel i5 Motherboard Block Diagram, Peripheral Connectors Pin Diagrams and functioning of I/O Hub, DDR-3 memory BUS

Student will submit the term work in the form of Journal consisting of minimum of 13 experiments with all seven experiments from group A and any 5 assignments from group B and group C assignments. Practical examination will be based on the term work and questions will be asked to judge the understanding of assignments performed at the time of examination.

210249 Computer Graphics and Gaming

Teaching Scheme
Lectures: 3 Hrs/week
Practical: 2Hrs/Week

Examination Scheme
Theory: 50 Marks
OnLine: 50 Marks
Term Work: 25 Marks

Prerequisite: Knowledge of C/C++ Programming and Basic Data Structures & Mathematics

Course Objectives
1. To understand basics of computer graphics
2. To give more emphasis on implementation aspect of Computer Graphics Algorithm.
3. To prepare the student for advance courses like multimedia / Computer Vision.

Course Outcome
3.Become knowledgeable about computer graphics hardware and peripherals
4.To generate 2D line and polygonal diagram and apply transformations.

Teaching aid
Faculties should use LCD to demonstrate the concept of Graphics.

Unit I	**Basic Concepts and devices**	**(6 Hrs)**
	Introduction to computer graphics, state of art applications of computer graphics, pixel, frame buffer, resolution, aspect ratio. Video display devices: CRT (Raster scan and random scan displays), flat panel displays. Interactive devices: joysticks, touch panels, light pens. Data generating devices: scanners and digitizers. graphics Files: TIFF. Introduction to GTK+.	
Unit II	**Scan Conversions**	**(6 Hrs)**
	Line and line segments, line and circle drawing algorithms: DDA and Bresenham , Line styles: thick, dotted and dashed. Antialising and antialising techniques. Character generating methods: stroke and bitmap method, Multiligual character standards. Concurrent line drawing using midpoint sub-division algorithm.	
Unit III	**Clipping and Windowing**	
	Polygon and Clipping Introduction to polygon, types: convex, concave and complex. Representation of polygon, Inside test, polygon filling algorithms – seed fill, scan line fill and filling with patterns. **Windowing and clipping**: viewing transformations, 2-D clipping: Cohen – Sutherland algorithm. Polygon clipping: Sutherland Hodgeman algorithm, generalized clipping	**(6 Hrs)**
Unit IV	**Geometric Transformations**	
	2-D transformations: introduction, matrices, Translation, scaling, rotation, homogeneous coordinates and matrix representation, translation, coordinate transformation, rotation about an arbitrary point, inverse and shear transformation. 3-D transformations: introduction, 3-D geometry, primitives, 3-D transformations and matrix representation, rotation about an arbitrary axis, concept of parallel and perspective projections, viewing parameters, 3-D clipping, 3-D viewing transformations.	**(6 Hrs)**
Unit V	**Curves, Fractals, Hidden Surfaces, Light and Color Models**	

	Hidden surfaces: introduction, back-face removal algorithm: Painter's algorithm, binary space partition, Warnock algorithm, Z –buffer. Light and Color: Introduction, Diffused illumination, point source illumination, Shading Algorithms, reflections, shadows, ray tracing, Transparancy, Color models and tables: RGB, HIS, Introduction to curve generation, interpolation, B-splines, Bezier curve, Blending function, fractals, Fractal lines and surfaces, Irregular Curves: Joints and Knots, Connectivity.	**(6 Hrs)**
Unit VI	**Getting Started with Animation and Gaming**	
	Segments: Introduction, segment table, segment creation, deletion, renaming. Image transformations, raster techniques. Animation: Conventional and computer based animation, Methods of Controlling Animations, Basic guidelines of animation, Gaming Technologies, animation languages: Introduction to OpenGL ES, Gaming platforms: NVIDIA workstation (Block Diagram only), i860(Block Diagram only), Graphics Memory Pipeline (Block diagram only). **Introduction to Interactive Graphics & usage of at least two tools of computer graphics (3D studio, Maya, or Similar open source tools)**	**(6 Hrs)**

Text Books:

1. S. Harrington, "Computer Graphics", 2nd Edition, McGraw-Hill Publications, 1987, ISBN 0 – 07 – 100472 – 6.
2. D. Rogers, "Procedural Elements for Computer Graphics", 2nd Edition, Tata McGraw-Hill Publication, 2001, ISBN 0 – 07 – 047371 – 4.

Reference Books:

1. J. Foley, V. Dam, S. Feiner, J. Hughes, "Computer Graphics Principles and Practice", 2nd Edition, Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.
2. D. Hearn, M. Baker, "Computer Graphics – C Version", 2nd Edition, Pearson Education, 2002, ISBN 81 – 7808 – 794 – 4.
3. D. Rogers, J. Adams, "Mathematical Elements for Computer Graphics", 2nd Edition, Tata McGraw-Hill Publication, 2002, ISBN 0 – 07 – 048677 – 8.
4. Beginning Android 4 Games Development, Mario Zechner, Robert Green, Apress, ISBN: 978-81-322-0575-3

210250 COMPUTER ORGANIZATIONS

Teaching Scheme
Lectures: 3 Hrs/week
Practical: 2Hrs/Week

Examination Scheme
Theory: 50 Marks
OnLine: 50 Marks

Pre requisites: Microprocessor Architecture

Course Objectives

1. To understand the structure, function and characteristics of computer systems
2. To understand the design of the various functional units of digital computers
3. To learn basics of Parallel Computer Architecture.

Course Outcome

1. Able to understand Von Neumann and dataflow Architecture block diagrams
2. Able to understand Internal block diagram and functioning of CPU, ALU and memory organization

UNIT I	**The evolution of computers and number operations**	**8 Hrs**
	Mechanical Era, Electronic computer, VLSI – Integrated circuits. **SOC** Processor architecture performance consideration performance measure speedup techniques. System Architectures – Microprocessor, Micro controller and parallel processing. Designing for Performance, Von-Neumann Architecture, Data flow architecture, Computer Components, Interconnection Structures, Bus Interconnection, Scalar Data Types, Fixed and Floating point numbers, IEEE 488 Number representation, Signed numbers, Integer Arithmetic, 2's Complement method for multiplication, Booths Algorithm,	
UNIT II	**Processor Organization**	**6 Hrs**
	Processor Basics: CPU organization, CPU Bus Organization: Central BUS, Buses on periphery, Additional features: RISC & CISC types representative commercial microprocessor of RISC & CISC types, Co-processors Data representation –Integer and floating point representation, Instruction set –Addressing modes formats, Machine Instruction characteristics, types of operands, types of operations, Instruction formats, Processor organization, Register Organization in 8086/88, 80386Dx and i7 microprocessors,	
UNIT III	**Data Paths and ALU**	**6 Hrs**
	Data Paths: Fixed point and floating point Arithmetic, ALU, Pipeline processing Case study of Intel Nehalem organization, pipelined and non-pipelined machine cycles	
UNIT IV	**Control Design Organization**	**5 Hrs**
	Control Design: Basic concepts Hardwired and micro-programmed control, Pipeline control, Example of ADD Instruction macro/micro design,	
UNIT V	**Memory and I/O Organization**	**6 Hrs**
	Memory systems, DDR3 Memory Organization, NUMA and UMA, caches memory mapped I/O and I/O mapped I/O, DMA, buses and standard interfaces –serial parallel buses –PCI, SCSI USB. USB bus organization to interface display and Printer, Case Study: Intel Nehalem Memory Organization.	

UNIT VI	**Advanced Computer Organizations**	**6 Hrs**
	Advanced computer Organizations (Block Diagrams only) The AMD Multicore Opteron, The Sun UltraSparc T1, The IBM Cell Broadband Engine (CBE), The Intel IA-64, The IA-64 model: Explicitly Parallel Instruction Computing, Prediction, Speculative Loads. 64-bit architectures i5/ i7 Desktop version and mobile version, NVDIA GPU architecture (Block diagram only).	

Text Books:

1. Computer Architecture and Organization, Jhon P Hays, 3nd Edition, McGraw-Hill Publication, 2001, ISBN 0071004793
2. Zaky S, Hamacher, "Computer Organization", 5nd Edition, McGraw-Hill Publications, 2001, ISBN 0071122184.

Reference Books:

1. Intel Microprocessor Hand book (or PDF File from www.microsoft.com)
2. Computer Organization by Tanunbum

210251 PROGRAMMING LABORATORY

Teaching Scheme
Lectures: -----
Practical: 2Hrs/Week

Examination Scheme
Term Work: 50 Marks
Oral : 50 Marks

Pre requisites: Microprocessor Architecture

Learning Objectives

4. To understand the structure, function and characteristics of computer systems
5. To understand the design of the various functional units of digital computers
6. To learn basics of Parallel Computer Architecture.

List of Practical Assignments:

Tools		
1.	Operating Systems (64-Bit)	Latest 64-BIT Version and update of Microsoft Windows 7/ Windows 8 Operating System onwards or 64-bit Open source Linux or its derivative, 32/64 bit Android 4 (for mobile) or above.
2.	Programming Tools (64-Bit)	MASM64x or equivalent, Microsoft Visual Studio x64 Intrinsics with IDE (Refer your MSDN copy or http://msdn.Microsoft.com OpenGL ES, GTK++, TC++, Cuda C++

A>	**Compulsory assignments**
1.	Writing a C/C++ Program to emulate CPU Architecture (Central Bus) Develop register, ALU level GUI to display results..
2.	Writing a C++ class for displaying pixel or point on the screen.
3.	Write a C++ class for a Line drawing method using overloading DDA and Bresenham's Algorithms, inheriting the pixel or point.
4	Write a C++ class for a circle drawing inheriting line class
B>	**At least Six (Use of Morphism, Inheritance and associated OO-programming in the implementation is expected)**
1	Write a program using python to draw a line with line styles (Thick, Thin, Dotted)
2.	Write a program in C/C++ to draw a line with line style (Thick, Thin, Dotted)
3	Write a program in C/C++ to draw a circle of desired radius.
4	Write a C/C++ program to draw a convex polygons (Square, Rectangle, Triangle)
5.	Write a C/C++ program to draw a Convex polygon with programmable edges.
6.	Write a C/C++ program to fill polygon using scan line algorithm
7.	Write a Java/ Python program to draw a line with line style (Thick, Thin, Dotted)
8	Write a program in to draw a circle of desired radius using VC++ 12 or above. Use of BITBLT command is expected.
9	Write a Java/Python program to draw a simple polygons (Square, Rectangle, Triangle)
10	Write a Java/Python program to draw a simple polygon with programmable edges, a. using mouse click event b. using dialog box to accept to accept edges ordered list and its size in pixels
11	Write a Java/Python program to fill polygon using scan line algorithm
12.	Write a program in C++ to test that given point is inside the polygon
13	Write a program in C++ draw a concave polygon
14	Write a program to scan fill the given concave polygon for Android Mobile Programming. Use Android ADT for Eclipse.
15	Write a class to implement the Booths Multiplier for 8/16/32/64-bit numbers using sign extended multiplication.
16	Write a C++ class for a Line drawing method using overloading DDA and Bresenham's

	Algorithms, inheriting the pixel or point. Use Android ADT for Eclipse.
C>	**At least One Advanced Technology Programming**
1	Use OpenGL ES to draw a line for Android Mobile
2	Use Microsoft IDE to Draw a line Diagram
3.	Use VRML to draw a line Diagram
4	Use Direct3D/Maya or open source equivalent to draw a Bouncing ball animation
5	Use Parallel programming using Cuda to draw polygoan.

Examination will be conducted on experiments performed