# Nagindas Khandwala College

**Revised Syllabus**
**And**
**Question Paper Pattern**
**Of Course**
**Of**
**Bachelor of Science Information Technology**
**(BSC IT) Programme**

**(Department Of IT) First Year**
***Semester II***

**Under Autonomy**

***(To be implemented from Academic Year- 2017-2018)***

**Bachelor of Computer Science (BSC CS) Program**

*Under Choice Based Credit, Grading and Semester System*

*Course Structure*

## FYBSC IT

*(To be implemented from Academic Year- 2017-2018)*

| FYBSC CS – SEMESTER II | | | | | | | |
|---|---|---|---|---|---|---|---|
| Course Code | Course | Hrs. of Instruction/Week | Exam Duration (Hours) | Maximum Marks | | | Credits |
| | | | | CIE | SEE | Total | |
| 1721UITOP | **Core Subject:** Object Oriented Programming | 4 | 2 1/2 Hours | 25 | 75 | 100 | 3 |
| 1722UITMA | **Core Subject:** Microprocessor Architecture | 4 | 2 1/2 Hours | 25 | 75 | 100 | 3 |
| 1723UITWP | **Core Subject:** Web Programming | 4 | 2 1/2 Hours | 25 | 75 | 100 | 3 |
| 1724UITGC | **Core Subject:** Green Computing | 4 | 2 1/2 Hours | 25 | 75 | 100 | 3 |
| 1725UITDM | **Allied Subject :** Numerical and Statistical Methods | 4 | 2 1/2 Hours | 25 | 75 | 100 | 3 |
| 1721UITPR | Core Subject Practical: Object Oriented Programming | 2 | 2 Hours | | 50 | 50 | 1 |
| 1722UITPR | Core Subject Practical: Microprocessor | 2 | 2 Hours | | 50 | 50 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Architecture | | | | | | |
| **1723UITPR** | Core Subject Practical:<br>Web Programming | **2** | **2 Hours** | | **50** | **50** | **1** |
| **1724UITPR** | Core Subject Practical:<br>Green Computing | **2** | **2 Hours** | | **50** | **50** | **1** |
| **1725UITPR** | Allied Subject Practical:<br>Numerical and Statistical Methods | **2** | **2 Hours** | | **50** | | **1** |
| | **TOTAL** | **30** | | | | | **20** |

| **Course Code** | **Course** | **Hrs. of Instruction/ week** | **Exam Duration (Hours)** | **Maximum Marks** | | | **Credits** |
|---|---|---|---|---|---|---|---|
| | | | | **CIE** | **SEE** | **Total** | |
| **1721UITOP** | **Part 1:**<br>**Object Oriented Programming** | 4 | 2 ½ hrs | 25 | 75 | 100 | 3 |

| **Sr. No.** | **Modules / Units** |
|---|---|
| **1** | **UNIT 1** |
| | Introduction to C++: input/output operations, sequential execution, decision making, looping<br>Arrays, functions, pointers, exception handling<br>(Implementation of the concepts from semester1 in C++) |
| **2** | **UNIT 2** |
| | **Principles of OOPS**: OOPS Paradigm, Basic Concepts of OOPS: Objects, Classes, Data Abstraction and Data Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Passing<br><br>**Classes and Objects**: Simple classes (Class specification, class members accessing), Defining member functions, passing object as an argument, Returning object from functions, friend classes, Pointer to object, Array of pointer to object. |

| | |
|---|---|
| **3** | **UNIT 3** |
| | **Constructors and Destructors**: Introduction, Default Constructor, Parameterized Constructor and examples, Destructors<br><br>**Polymorphism**: Concept of function overloading, overloaded operators, overloading unary and binary operators, overloading comparison operator, overloading arithmetic assignment operator, Data Conversion between objects and basic types, |
| **4** | **UNIT 4** |
| | **Virtual Functions**: Introduction and need, Pure Virtual Functions, Static Functions, this Pointer, abstract classes, virtual destructors.<br><br>**Program development using Inheritance**: Introduction, understanding inheritance, Advantages provided by inheritance, choosing the access specifier, Derived class declaration, derived class constructors, class hierarchies |
| 5 | **UNIT 5** |
| | **Inheritance:** multiple inheritance, multilevel inheritance, containership, hybrid inheritance.<br><br>**Templates**: Introduction, Function Template and examples, Class Template and examples.<br><br>**Working with Files**: Introduction, File Operations, Various File Modes, File Pointer and their Manipulation |

| **Reference Books** |
|---|
| Object Oriented Programming |
| **References:**<br><br>1. Timothy Budd, Object Oriented Analysis and Design, TMH, 3rd Ed., 2012<br>2. K. R. Venugopal, Rajkumar Buyya, T. Ravishankar, Mastering C++, Tata McGraw Hill, 2nd Ed., 2011<br>3. Scott Meyers, Effective Modern C++, SPD<br>4. B. M. Hirwani, C++ for Beginners, SPD, 2013 |

## Practical (1721UITPR)

1. Classes and methods
   a) Design an employee class for reading and displaying the employee information, the getInfo() and displayInfo() methods will be used repectively. Where getInfo() will be private method
   b) Design the class student containing getData() and displayData() as two of its methods which will be used for reading and displaying the student information respectively.Where getData() will be private method.
   c) Design the class Demo which will contain the following methods: readNo(), factorial() for calculating the factorial of a number, reverseNo() will reverse the given number, isPalindrome() will check the given number is palindrome, isArmstrong() which will calculate the given number is armStrong or not.Where readNo() will be private method.
   d) Write a program to demonstrate function definition outside class and accessing class members in function definition.
2. Using friend functions.
   a) Write a friend function for adding the two complex numbers, using a single class
   b) Write a friend function for adding the two different distances and display its sum, using two classes.
   c) Write a friend function for adding the two matrix from two different classes and display its sum
3. Constructors and method overloading.
   a) Design a class Complex for adding the two complex numbers and also show the use of constructor.
   b) Design a class Geometry containing the methods area() and volume() and also overload the area() function .
   c) Design a class StaticDemo to show the implementation of static variable and static Function
4. Operator Overloading
   a) Overload the operator unary(-) for demonstrating operator overloading.
   b) Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.
   c) Overload the + for concatenating the two strings. For e.g “Py” + “thon” = Python
5. Inheritance
   a) Design a class for single level inheritance using public and private type derivation
   b) Design a class for multiple inheritance
   c) Implement the hierarchical inheritance
6. Virtual functions and abstract classes
   a) Implement the concept of method overriding, virtual function, abstract class
7. String handling
   a) String operations for string length , string concatenation, string reverse, string comparison
   b) Console formatting functions
8. Exception handling
   a) Show the implementation of exception handling, exception handling for strings
   b) Show the implementation of exception handling for using the pointers
9. File handling
   a) Design a class FileDemo open a file in read mode and display the total number of
   b) words and lines in the file.
   c) Design a class to handle multiple files and file operations
   d) Design a editor for appending and editing the files
10. Templates
   a) Show the implementation of template class library for swap function.
   b) Design the template class library for sorting ascending to descending and vice-versa

| Course Code: | Course | Hrs. of Instruction/ week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| 1722UITMA | Microprocessor Architecture | 4 | 2 ½ hrs | 25 | 75 | 100 | 3 |

| Sr. No. | Modules / Units |
|---|---|
| 1 | **UNIT 1** |
| | **Microprocessor, microcomputers, and Assembly Language**:<br><br>Microprocessor, Microprocessor Instruction Set and Computer Languages, From Large Computers to Single-Chip Microcontrollers, Applications.<br><br>**Microprocessor Architecture and Microcomputer System**:<br><br>Microprocessor Architecture and its operation's, Memory, I/O Devices, Microcomputer System, LogicDevices and Interfacing, Microprocessor-Based System Application.<br><br>**8085 Microprocessor Architecture and Memory Interface**:<br><br>Introduction, 8085 Microprocessor unit, 8085-Based Microcomputer, Memory Interfacing, Interfacing the 8155 Memory Segment, Illustrative Example: Designing Memory for the MCTS Project, Testing and Troubleshooting Memory Interfacing Circuit, 8085-Based Single-Board microcomputer. |
| 2 | **UNIT 2** |
| | **Interfacing of I/O Devices**<br><br>Basic Interfacing concepts, Interfacing Output Displays, Interfacing Input Devices, Memory Mapped I/O, Testing and Troubleshooting I/O Interfacing Circuits.<br><br>**Introduction to 8085 Assembly Language Programming**:<br><br>The 8085 Programming Model, Instruction Classification, Instruction, Data and Storage, Writing assembling and Execution of a simple program, Overview of 8085 Instruction Set, Writing and Assembling Program.<br><br>**Introduction to 8085 Instructions:**<br><br>Data Transfer Operations, Arithmetic Operations, Logic Operation, Branch Operation, Writing Assembly Languages Programs, Debugging a Program. |

| | |
|---|---|
| **3** | **UNIT 3** |
| | **Programming Techniques With Additional Instructions**:<br><br>Programming Techniques: Looping, Counting and Indexing, Additional Data Transfer and 16-Bit Arithmetic Instructions, Arithmetic Instruction Related to Memory, Logic Operations: Rotate, Logics Operations: Compare, Dynamic Debugging.<br><br>**Counters and Time Delays**:<br><br>Counters and Time Delays, Illustrative Program: Hexadecimal Counter, Illustrative Program: zero-to-nine (Modulo Ten) Counter, Generating Pulse Waveforms, Debugging Counter and Time-Delay Programs.<br><br>**Stacks and Sub-Routines**:<br><br>Stack, Subroutine, Restart, Conditional Call, Return Instructions, Advanced Subroutine concepts. |
| **4** | **UNIT 4** |
| | **Software Development System and Assemblers**:<br><br>Microprocessors-Based Software Development system, Operating System and Programming Tools, Assemblers and Cross-Assemblers, Writing Program Using Cross Assemblers.<br><br>**Interrupts**:<br><br>The 8085 Interrupt, 8085 Vectored Interrupts, Restart as S/W Instructions, Additional I/O Concepts and processes |
| 5 | **UNIT 5** |
| | **The Pentium and Pentium Pro microprocessors**: Introduction, Special Pentium registers, Memory management, Pentium instructions, Pentium Pro microprocessor, Special Pentium Pro features.<br><br>**Core 2 and later Microprocessors**: Introduction, Pentium II software changes, Pentium IV and Core 2, i3, i5 and i7.<br><br>**SUN SPARC Microprocessor**: Architecture, Register file, data types and instruction format |

## Reference Books

### Microprocessor Architecture

**References:**


1. Ramesh Gaonkar, Microprocessor Architecture, programming and Applications with the 8085, PENRAM, 5th Ed., 2012
2. M. Morris Mano, Computer System Architecture, PHI, 1998
3. Andrew C. Tanenbaum, Structured Computer organization, PHI

## Practical (1722UITPR)

1. Perform the following Operations related to memory locations
   a. Store the data byte 32H into memory location 4000H.
   b. Exchange the contents of memory locations 2000H and 4000H
2. Simple assembly language programs.
   a. Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.

   b. Subtract two 8-bit numbers.

   c. Add the 16-bit number in memory locations 4000H and 4001H to the 16-bit number in memory locations 4002H and 4003H. The most significant eight bits of the two numbers to be added are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.

   d. Add the contents of memory locations 40001H and 4001H and place the result in the memory locations 4002Hand 4003H.

   e. Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.
   f. Find the l's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H.
   g. Find the 2's complement of the number stored at memory location 4200H and store the complemented number at memory location 4300H.
3. Packing and unpacking operations.

   a. Pack the two unpacked BCD numbers stored in memory locations 4200H and 4201H and store result in memory location 4300H. Assume the least significant digit is stored at 4200H.

   b. Two digit BCD number is stored in memory location 4200H. Unpack the BCD number and store the two digits in memory locations 4300H and 4301H such that memory location 4300H will have lower BCD digit.

4. Register Operations.

   a. Write a program to shift an eight bit data four bits right. Assume that data is in register C.

   b. Program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair

   c. Write a set of instructions to alter the contents of flag register in 8085.

   d. Write a program to count number of l's in the contents of D register and store the count in the B register.
5. Multiple memory locations.

1. Calculate the sum of series of numbers. The length of the series is in memory location 4200H and the series begins from memory location 4201H. a. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memory location 4300H. b. Consider the sum to be 16 bit number. Store the sum at memory locations 4300H and 4301H

2. Multiply two 8-bit numbers stored in memory locations 2200H and 2201H by repetitive addition

and store the result in memory locations 2300H and 2301H.

3. Divide 16 bit number stored in memory locations 2200H and 2201H by the 8 bit number stored at memory location 2202H. Store the quotient in memory locations 2300H and 2301H and remainder in memory locations 2302H and 2303H.

4. Find the number of negative elements (most significant bit 1) in a block of data. The length of the block is in memory location 2200H and the block itself begins in memory location 2201H. Store the number of negative elements in memory location 2300H

5. Find the largest number in a block of data. The length of the block is in memory location 2200H and the block itself starts from memory location 2201H. Store the maximum number in memory location 2300H. Assume that the numbers in the block are all 8 bit unsigned binary numbers.

6. Calculations with respect to memory locations.

a. Write a program to sort given 10 numbers from memory location 2200H in the ascending order.

b. Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location 2Sample problem:

c. Calculate the sum of series of odd numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 16-bit. Store the sum at memory locations 2300H and 2301H.

d. Find the square of the given numbers from memory location 6100H and store the result from memory location 7000H

e. Search the given byte in the list of 50 numbers stored in the consecutive memory locations and store the address of memory location in the memory locations 2200H and 2201H. Assume byte is in the C register and starting address of the list is 2000H. If byte is not found store 00 at 2200H and 2201H

f. Two decimal numbers six digits each, are stored in BCD package form. Each number occupies a sequence of byte in the memory. The starting address of first number is 6000H Write an assembly language program that adds these two numbers and stores the sum in the same format starting from memory location 6200H

7. Add 2 arrays having ten 8-bit numbers each and generate a third array of result. It is necessary to add the first element of array 1 with the first element of array-2 and so on. The starting addresses of array l, array2 and array3 are 2200H, 2300H and 2400H, respectively
Assembly programs on memory locations.

a. Write an assembly language program to separate even numbers from the given list of 50 numbers and store them in the another list starting from 2300H. Assume starting address of 50 number list is 2200H

b. Write assembly language program with proper comments for the following:

c. A block of data consisting of 256 bytes is stored in memory starting at 3000H. This block is to be shifted (relocated) in memory from 3050H onwards. Do not shift the block or part of the block anywhere else in the memory.

d. Add even parity to a string of 7-bit ASCII characters. The length of the string is in memory location

2040H and the string itself begins in memory location 2041H. Place even parity in the most significant bit of each character.

e. A list of 50 numbers is stored in memory, starting at 6000H. Find number of negative, zero and positive numbers from this list and store these results in memory locations 7000H, 7001H, and 7002H respectively

f. Write an assembly language program to generate fibonacci number.

g. Program to calculate the factorial of a number between 0 to 8.

8. String operations in assembly programs.

a. Write an 8085 assembly language program to insert a string of four characters from the tenth location in the given array of 50 characters

b. Write an 8085 assembly language program to delete a string of 4 characters from the tenth location in the given array of 50 characters.

c. Multiply the 8-bit unsigned number in memory location 2200H by the 8-bit unsigned number in memory location 2201H. Store the 8 least significant bits of the result in memory location 2300H and the 8 most significant bits in memory location 2301H.

d. Divide the 16-bit unsigned number in memory locations 2200H and 2201H (most significant bits in 2201H) by the B-bit unsigned number in memory location 2300H store the quotient in memory location 2400H and remainder in 2401H

e. DAA instruction is not present. Write a sub routine which will perform the same task as DAA.

9. Calculations on memory locations.

a. To test RAM by writing '1' and reading it back and later writing '0' (zero) and reading it back. RAM addresses to be checked are 40FFH to 40FFH. In case of any error, it is indicated by writing 01H at port 10

b. Arrange an array of 8 bit unsigned no in descending order

c. Transfer ten bytes of data from one memory to another memory block. Source memory block starts from memory location 2200H where as destination memory block starts from memory location 2300H

d. Write a program to find the Square Root of an 8 bit binary number. The binary number is stored in memory location 4200H and store the square root in 4201H.

e. Write a simple program to Split a HEX data into two nibbles and store it in memory

10. Operations on BCD numbers.

a. Add two 4 digit BCD numbers in HL and DE register pairs and store result in memory locations, 2300H and 2301H. Ignore carry after 16 bit.

b. Subtract the BCD number stored in E register from the number stored in the D register

c. Write an assembly language program to multiply 2 BCD numbers

| Course Code | Course | Hrs. of Instruction/Week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| 1723UITWP | Core 1: Web Programming | 4 | 2 1/2 Hours | 25 | 75 | 100 | 3 |

| Sr. No. | Modules / Units |
|---|---|
| 1 | **UNIT 1** |
| | **Internet and the World Wide Web:**<br><br>What is Internet? Introduction to internet and its applications, E-mail, telnet, FTP, e-commerce, video conferencing, e-business. Internet service providers, domain name server, internet address, World Wide Web (WWW): World Wide Web and its evolution, uniform resource locator (URL), browsers – internet explorer, Netscape navigator, opera, Firefox, chrome, Mozilla. search engine, web saver – apache, IIS, proxy server, HTTP protocol<br><br>**HTML5:**<br><br>Introduction, Why HTML5? Formatting text by using tags, using lists and backgrounds, Creating hyperlinks and anchors. Style sheets, CSS formatting text using style sheets, formatting paragraphs using style sheets. |
| 2 | **UNIT 2** |
| | **HTML5 Page layout and navigation:**<br>Creating navigational aids: planning site organization, creating text based navigation bar, creating graphics based navigation bar, creating graphical navigation bar, creating image map, redirecting to another URL, creating division based layouts: HTML5 semantic tags, creating divisions, creating HTML5 semantic layout, positioning and formatting divisions.<br>**HTML5 Tables, Forms and Media: Creating tables:**<br>creating simple table, specifying the size of the table, specifying the width of the column, merging table cells, using tables for page layout, formatting tables: applying table borders, applying background and foreground fills, changing cell padding, spacing and alignment, creating user forms: creating basic form, using check boxes and option buttons, creating lists, additional input types in HTML5, Incorporating sound and video: audio and video in HTML5, HTML multimedia basics, embedding video clips, incorporating audio on web page. |

| | |
|---|---|
| **3** | **UNIT 3** |
| | **Java Script:**<br><br>Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security,<br><br>**Operators:**<br><br>Assignment Operators, Comparison Operators, Arithmetic Operators, % (Modulus), ++(Increment), --(Decrement), -(Unary Negation), Logical Operators, Short-Circuit Evaluation, String Operators, Special Operators, ?: (Conditional operator), , (Comma operator), delete, new, this, void<br><br>**Statements:**<br><br>Break, comment, continue, delete, do...while, export, for, for...in, function, if...else, import, labelled, return, switch, var, while, with,<br><br>Core JavaScript (Properties and Methods of Each) : Array, Boolean, Date, Function, Math, Number, Object, String, regExp Document and its associated objects: document, Link, Area, Anchor, Image, Applet, Layer<br><br>**Events and Event Handlers** : General Information about Events, Defining Event Handlers, event, onAbort, onBlur, onChange, onClick, onDblClick, onDragDrop, onError, onFocus, onKeyDown, onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove, onReset, onResize, onSelect, onSubmit, onUnload |
| **4** | **UNIT 4** |
| | **PHP:**<br><br>Why PHP and MySQL? Server-side scripting, PHP syntax and variables, comments, types, control structures, branching, looping, termination, functions, passing information with PHP, GET, POST, formatting form variables, superglobal arrays, strings and string functions, regular expressions, arrays, number handling, basic PHP errors/problems |
| 5 | **UNIT 5** |
| | **Advanced PHP and MySQL :**<br><br>PHP/MySQL Functions, Integrating web forms and databases, Displaying queries in tables, Building Forms from queries, String and Regular Expressions, Sessions, Cookies and HTTP, E-Mail<br><br>Introduction to Bootstrap (elementary level) – Introduction to JQuery |

| Reference Books |
| --- |
| **Web Programming** |

**References**:


1. Thomas Powell, Web Design: The Complete Reference, Tata McGraw Hill
2. Faithe Wempen, HTML5 Step by Step, Microsoft Press, 2011
3. Ivan Bayross, Sharanam Sha, PHP 5.1 for Beginners, SPD, 2013
4. Sharanam Shah, Vaishali Shah, , PHP Project for Beginners, SPD, 2015
5. Steve Suehring, Tim Converse, Joyce Park, PHP 6 and MySQL Bible, Wiley, 2009
6. Eric Freeman, Head First HTML 5 Programming, O'Reilly, 2013
7. Thomas Powell and Fritz Schneider, JavaScript 2.0: The Complete Reference, Tata McGraw Hill, 2nd Ed.


| Practical (1723UITPR) |
| --- |
| 1. Use of Basic Tags<br>a) Design a web page using different text formatting tags.<br>b) Design a web page with links to different pages and allow navigation between web pages<br>c) Design a web page demonstrating all Style sheet types<br>2. Use of Table tags, attributes and style properties<br>a) Design a simple table using border & border collapse property<br>b) Design a table with merge cells<br>c) Design a table illustrating cell padding, cell spacing & different border styles<br>d) Design a table illustrating text alignments<br>3. Image maps, Forms and Media<br>a) Design a web page with Imagemaps.<br>b) Design a web page demonstrating different semantics<br>c) Design a web page with different tables. Design a webpages using table so that the content appears well placed<br>d) Design a web page with a form that uses all types of controls.<br>e) Design a web page embedding with multimedia features<br>4. Java Script<br>a) Using JavaScript design, a web page that prints factorial/Fibonacci series/any given series<br>b) Design a form and validate all the controls placed on the form using Java Script.<br>c) Write a JavaScript program to display all the prime numbers between 1 and 100.<br>d) Write a JavaScript program to accept a number from the user and display the sum of its digits.<br>e) Write a program in JavaScript to accept a sentence from the user and display the number of words in it. (Do not use split () function).<br>f) Write a java script program to design simple calculator.<br>5. Control and looping statements and Java Script references<br>a) Design a web page demonstrating different conditional statements.<br>b) Design a web page demonstrating different looping statements.<br>c) Design a web page demonstrating different Core JavaScript references (Array, Boolean, Date, Function, Math, Number, Object, String, regExp).<br>6. Basic PHP I<br>a) Write a PHP Program to accept a number from the user and print it factorial<br>b) Write a PHP program to accept a number from the user and print whether it is prime or not. |

7. Basic PHP II

a) Write a PHP code to find the greater of 2 numbers. Accept the no. from the user.

b) Write a PHP program to display the following Binary Pyramid:

```
1
0 1
1 0 1
0 1 0 1
```

8. String Functions and arrays

a) Write a PHP program to demonstrate different string functions.

b) Write a PHP program to create one dimensional array.

9. PHP and Database

a) Write a PHP code to create:

b) Create a database College

c) Create a table Department (Dname, Dno, Number_Of_faculty)

d) Write a PHP program to create a database named "College". Create a table named "Student" with following fields (sno, sname, percentage). Insert 3 records of your choice. Display the names of the students whose percentage is between 35 to 75 in a tabular format.

e) Design a PHP page for authenticating a user.

10. Email, Sessions and Cookies

a) Write a program to send email with attachment.

b) Write a program to demonstrate use of sessions and cookies.

| Course Code: | Course | Hrs. of Instruction/ week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| 1724UITGC | DSE 1: Green Computing | 4 | 2 ½ hrs | 25 | 75 | 100 | 3 |

| Sr. No. | Modules / Units |
|---|---|
| 1 | **UNIT 1** |
| | **FUNDAMENTALS of GREEN IT**<br><br>Green IT Fundamentals, Problems: Toxins, Power Consumption, Equipment Disposal , Green computing: carbon foot print, scoop on power, Green IT Strategies: Drivers, Dimensions, and Goals, Environmentally Responsible Business: Policies, Practices, and Metrics.<br><br>**Governance and Regulatory Issues**<br>Global Initiatives: United Nations, Basel Action Network, Basel Convention, North America: The United States, Canada, Australia, Europe, WEEE Directive, RoHS, National Adoption, Asia: Japan, China, Korea. |
| 2 | **UNIT 2** |
| | **Minimizing Power Usage**:<br>Power Problems, Monitoring Power Usage, Servers, Low-Cost Options, Reducing Power Use, Data De-Duplication, Virtualization, Management, Bigger Drives, Involving the Utility Company, Low-Power Computers, PCs, Linux, Components, Servers, Computer Settings, Storage, Monitors, Power Supplies, Wireless Devices, Software.<br>**Cooling**:<br>Cooling Costs, Power Cost, Causes of Cost, Calculating Cooling Needs, Reducing Cooling Costs, Economizers, On-Demand Cooling, HP's Solution, Optimizing Airflow, Hot Aisle/Cold Aisle, Raised Floors, Cable Management, Vapour Seal, Prevent Recirculation of Equipment Exhaust, Supply Air Directly to Heat Sources, Fans, Humidity, Adding Cooling, Fluid Considerations, System Design, Datacentre Design, Centralized Control, Design for Your Needs, Put Everything Together. |
| 3 | **UNIT 3** |
| | **Grid Framework**:<br><br>Virtualizing of IT systems – Role of electric utilities, Telecommuting, teleconferencing and teleporting – Materials recycling – Best ways for Green PC – Green Data center – Green Grid framework.<br><br>**Going Paperless**:<br>Paper Problems, The Environment, Costs: Paper and Office, Practicality, Storage, Destruction, Going Paperless, Organizational Realities, Changing Over, Paperless |

| | |
|---|---|
| | Billing, Handheld Computers vs. the Clipboard, Unified Communications, Intranets, What to Include, Building an Intranet, Microsoft Office SharePoint Server 2007, Electronic Data Interchange (EDI), Nuts and Bolts, Value Added Networks, Advantages, Obstacles. |
| **4** | **UNIT 4** |
| | **Green Data Storage**: Introduction , Storage Media Power Characteristics, Energy Management Techniques for Hard Disks, System-Level Energy Management<br>Green Networks and Communications: Introduction, Objectives of Green Network Protocols, Green Network Protocols and Standards<br>**Recycling**:<br>Problems, China, Africa, Materials, Means of Disposal, Recycling, Refurbishing, Make the Decision, Life Cycle, from beginning to end,<br>Life, Cost, Green Design, Recycling Companies, Finding the Best One, Checklist, Certifications, Hard Drive Recycling, Consequences, cleaning a Hard Drive, Pros and cons of each method, CDs and DVDs, good and bad about CD and DVDs disposal, Change the mind-set, David vs. America Online |
| 5 | **UNIT 5** |
| | **Green Assets**<br><br>Greening Your Information Systems: Initial Improvement Calculations, Selecting Metrics, Tracking Progress, Change Business Processes, Customer Interaction, Paper Reduction, Green Supply Chain, Improve Technology Infrastructure, Reduce PCs and Servers, Shared Services, Hardware Costs, Cooling. Green Data Centres: Data Centres and Associated Energy Challenges, Data Centre IT Infrastructure, Data Centre Facility Infrastructure: Implications for Energy Efficiency, IT Infrastructure Management, Green Data Centre Metrics<br>**Staying Green**:<br>Organizational Check-ups, Chief Green Officer, Evolution, Sell the CEO, SMART Goals, Equipment Check-ups, Gather Data, Tracking the data, Baseline Data, Benchmarking, Analyse Data, Conduct Audits, Certifications, Benefits, Realities, Helpful Organizations |

## Reference Books

### Green Computing

**Reference books:**


1. Toby Velte, Anthony Velte, Robert Elsenpeter, Green IT, McGraw Hill, 2008
2. Alvin Galea, Michael Schaefer, Mike Ebbers, Green Data Center: Steps for the Journey, Shroff Publishers and Distributers, 2011
3. Jason Harris, Green Computing and Green IT Best Practice, Emereo
4. Bud E. Smith, Green Computing Tools and Techniques for Saving Energy, Money and Resources, CRC Press, 2014
5. San Murugesan, G. R. Ganadharan, Harnessing Green IT: Principles and Practices, Wiley & IEEE

| **Practical (1724UITPR)** |
| --- |
| 1. Project and Viva-Voce<br>A project should be done based on the objectives of Green Computing. A report of minimum 50 pages should be prepared. The report should have a font size of 12, Times new roman and 1.5 line spacing. The headings should have font size 14. The report should be hard bound.<br>➢ The project can be done individually or a group of two students.<br>➢ The students will have to present the project during the examination.<br>➢ A certified copy of the project report is essential to appear for the examination. |

| Course Code | Course | Hrs. of Instruction/Week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| 1725UITDM | Core 2: Numerical and Statistical methods | 4 | 2 1/2 Hours | 25 | 75 | 100 | 3 |

| Sr. No. | Modules / Units |
|---|---|
| 1 | **UNIT 1** |
| | **Mathematical Modeling and Engineering Problem Solving**: A Simple Mathematical Model, Conservation Laws and Engineering Problems<br>**Approximations and Round-Off Errors**: Significant Figures, Accuracy and Precision, Error Definitions, Round-Off Errors<br>**Truncation Errors and the Taylor Series**: The Taylor Series, Error Propagation, Total Numerical Errors, Formulation Errors and Data Uncertainty |
| 2 | **UNIT 2** |
| | **Solutions of Algebraic and Transcendental Equations**: The Bisection Method, The Newton-Raphson Method, The Regula-falsi method, The Secant Method.<br>**Interpolation**: Forward Difference, Backward Difference, Newton's Forward Difference Interpolation, Newton's Backward Difference Interpolation, Lagrange's Interpolation. |
| 3 | **UNIT 3** |
| | **Solution of simultaneous algebraic equations (linear) using iterative methods**: Gauss-Jordan Method, Gauss-Seidel Method.<br>**Numerical differentiation and Integration**:Numberical differentiation, Numerical integration using Trapezoidal Rule, Simpson's 1/3rd and 3/8th rules.<br><br>**Numerical solution of 1st and 2nd order differential equations**: Taylor series, Euler's Method, Modified Euler's Method, Runge-Kutta Method for 1st and 2nd Order Differential Equations. |
| 4 | **UNIT 4** |
| | **Least-Squares Regression**:<br>Linear Regression, Polynomial Regression, Multiple Linear Regression, General Linear Least Squares, Nonlinear Regression<br><br>**Linear Programming**: Linear optimization problem, Formulation and Graphical solution, Basic solution and Feasible solution. |

| 5 | **UNIT 5** |
|---|---|
| | **Random variables**: Discrete and Continuous random variables, Probability density function, Probability distribution of random variables, Expected value, Variance.<br><br>**Distributions**: Discrete distributions: Uniform, Binomial, Poisson, Bernoulli, **Continuous distributions**: uniform distributions, exponential, (derivation of mean and variance only and state other properties and discuss their applications) Normal distribution state all the properties and its applications. |

| **Reference Books** |
|---|
| **Numerical and Statistical Methods** |
| **References:**<br><br>1. S. S. Shastri, Introductory Methods of Numeriacal Methods, PHI, Vol.2<br>2. Steven C. Chapra, Raymond P. Canale, Numerical methods for Engineers, Tata McGraw Hill, 6th Ed., 2010<br>3. Richard L., Burden J, Douglas Faires, Numerical Analysis, Cangage Learning, 9th Ed., 2011<br>4. S. C. Gupta, V. K. Kapoor, Fundamentals of Mathematical Statistics,<br>5. P. N. Wartikar and J. N. Wartikar, Elements of Applied Mathematics, A. V. Griha, Pune, Vol. 1 & 2 |

## Practical (1725UITPR)

1. Iterative Calculation

a) Program for iterative calculation.
b) Program to calculate the roots of a quadratic equation using the formula.
c) Program to evaluate using infinite series.

2. Solution of algebraic and transcendental equations:

a) Program to solve algebraic and transcendental equation by bisection method.
b) Program to solve algebraic and transcendental equation by false position method.
c) Program to solve algebraic and transcendental equation by Secant method.
d) Program to solve algebraic and transcendental equation by Newton Raphson method.

3. Interpolation

a) Program for Newton's forward interpolation.
b) Program for Newton's backward interpolation.
c) Program for Lagrange's interpolation.

4. Solving linear system of equations by iterative methods

a) Program for solving linear system of equations using Gauss Jordan method.
b) Program for solving linear system of equations using Gauss Seidel method.

5. Numerical Differentiation

a) Programing to obtain derivatives numerically.

6. Numerical Integration

a) Program for numerical integration using Trapezoidal rule.
b) Program for numerical integration using Simpson's 1/3rd rule.
c) Program for numerical integration using Simpson's 3/8th rule.

7. Solution of differential equations

a) Program to solve differential equation using Euler's method
b) Program to solve differential equation using modified Euler's method.
c) Program to solve differential equation using Runge-kutta 2nd order and 4th order methods.

8. Regression

a) Program for Linear regression.
b) Program for Polynomial Regression.
c) Program for multiple linear regression.
d) Program for non-linear regression.

9. Random variables and distributions

a) Program to generate random variables.
b) Program to fit binomial distribution.
c) Program to fit Poisson distribution.

10. Distributions

a) Program for Uniform distribution.
b) Program for Bernoulli distribution
c) Program for Negative binomial distribution

# Evaluation Scheme

**I. Internal Exam-25 Marks**

**(i) Test– 20 Marks -** Duration 40 mins

It will be conducted either as a written test or using any open source learning management system such as Moodle (Modular object-oriented dynamic learning environment)Or a test based on an equivalent online course on the contents of the concerned course(subject)offered by or build using MOOC (Massive Open Online Course)platform.

**(ii) 5 Marks -** Active participation in routine class instructional deliveries

Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

**II. External Examination- 75 Marks**

**(i)** Duration - 2.5 Hours.

**(ii)** Theory question paper pattern:-

| All questions are compulsory. | | |
|---|---|---|
| Question | Based on | Marks |
| Q.1 | Unit 1 | 15 |
| Q.2 | Unit 2 | 15 |
| Q.3 | Unit 3 | 15 |
| Q.4 | Unit 4 | 15 |
| Q.5 | Unit 5 | 15 |

- All questions shall be compulsory with internal choice within the questions.
- Each Question may be sub-divided into sub questions as a, b, c, d & e, etc & the allocation of Marks depends on the weightage of the topic.

**III. Practical Examination – 50 marks (Duration: 2 Hours)**

- Each practical course carries 50 Marks : 40 marks + 05 marks (journal)+ 05 marks(viva)
- Minimum 75% practical from each core/allied course are required to be completed and written in the journal**.**

**(Certified Journal is compulsory for appearing at the time of Practical Exam)**