Shiksha Mandal's
Bajaj College of Science, Wardha
(Formerly known as Jankidevi Bajaj College of Science, Wardha)

Syllabus of

Bachelor of Science (B. Sc.) Three Year (Six Semesters) Degree Course

Semester I

Subject: Computer Science Major

(Approved in BoS meeting held on 15.05.2023 w.e.f. Academic Session2023-24)

Programme Objective:

There are two main objectives to the B.Sc. Computer Science Programme.

a) The primary goal of the degree program in Computer Science is to provide students the foundations for the future work and careers in computation-based courses. These foundations support both a successful career path in computing as well as provide appropriate qualifications for further degree work in computation related disciplines. Our degree emphasizes development of analytical skills, acquisition of knowledge and understanding of systems, languages and tools required for effective computation-based problem solving.

b) A Computer Science major will provide you with knowledge of programming, operating systems, compiler design and principles of programming language. These skills will prepare graduates to enter technological fields such as systems programming, technical support, research and teaching.

On completion of the B.Sc. Computer Science Programme, the students:

a) Will have sound knowledge of the Programming languages such as C, C++, Java, VB and SQL. Also they will have knowledge about Information Technology, Operating System, System Analysis and Design and Development.

b) Can participate in off-campus internships, independent study projects, part-time computer employment, lectures by guest speakers, Society of Computing Students programming and on campus employment as lab assistants, lab supervisors and Helpdesk assistants.

c) Be in a position to develop industrial and entrepreneur applications.

Aim of the Course

To understand the basics of computer hardware, software, network and algorithms, flowcharts and programming and also to provide sound knowledge of object oriented programming concepts and their understanding, development, working process of software.

Syllabus for B.Sc. SEMESTER I
(To be implemented from Academic Session 2023-24)

Learning Objectives	Course Learning Outcomes
Students will try to learn:	After successful completion of the course student will be able to
1. To understand the concepts of fundamentals of a computer system, number systems, generations of computer languages, and other system tools like translator programs.	1. Understand various number systems, generations of computer languages and translators.
2. To understand various types of memories, storage devices, input and output devices.	2. Know various types of memories, storage devices, input and output devices.
3. To get acquainted with concepts of networks and its associated terminologies, topologies, architecture, communication devices and recent technologies like Bluetooth and Wi-Fi.	3. Learn the concepts of networks, topologies, architecture, communication devices and recent technologies.
4. To understand the basics of 'C' programming. Also get knowledge of algorithm and flowcharts.	4. Learn to make basics programs and also able to draw algorithm and flowcharts of programs.
5. To know the concepts of control and looping structure. Also student get the idea about concepts of arrays and functions.	5. Perform programs using control and looping structure also understand the basic programs on arrays and functions.
6. To understand the basics of structures, pointers and file handling.	6. Implementation of structures, pointers and file handling concepts in programming.

Subject: Computer Science Major

DSC - I

FUNDAMENTALS OF INFORMATION TECHNOLOGY and PROGRAMMING IN 'C'

Course Code: UCS110T

Credits: 6 (4 Th, 2 Pr) **No. of Lectures: 60** **No. of Practical: 60**

Course Outline

Unit I

Evolution of Computers, Generations of Computer, Characteristics of a Computer, Computer Classification.

Basic Components of Digital Computers: Block Diagram. CPU: Functions of Each Unit: Primary Memory, ALU and CU, Instruction format. Bus: Data, Control and Address Bus. Number Systems: Binary, Octal, Decimal, Hexadecimal, Their Conversions, Binary Arithmetic. ASCII, BCD, EBCDIC.

Language Evolution: Generation of Languages: Machine, Assembly, High Level Languages. Characteristics of Good Language Translators: Compiler, Interpreter and Assembler. Source and object Program.

Unit II

Memory: Memory hierarchy, RAM, Static & dynamic RAM, Types of RAM (SDRAM, RDRAM, DDR), ROM, PROM, EPROM, EEPROM, Flash and Cache. Storage Devices: Hard Disk, Zip Disk and Optical Disk. Pen Drive, Blu Ray and SSD.

Input Devices: Keyboard, Mouse, Light Pen, Touch Screen, Voice Input, MICR, OCR, OMR, Barcode Reader and Flatbed Scanner. Output Devices: VDU, Printers: Dot Matrix, Laser and Inkjet. Plotters: Drum, Flat-Bed and Inkjet.

Unit III

Computer Network: Goals and applications - Business Application, Home Application, Network terminology, Topologies: Linear, Circular, Tree and Mesh. Types of Networks: LAN, WAN, MAN. Repeaters, Bridge, Routers, Brouters and Gateway. Modem for Communication between PC's, Wi-Fi network, Introduction of Bluetooth and Infrared devices. Network protocols. Architecture: Peer-to-Peer, Client/Server. TCP/IP Reference Model with functionality of each layer.

Internet services (Introduction only): WWW – Web browser, URL, Internet search engines, WWW development languages, Electronic mail – E-mail address, e-mail message format, e-mail services (application based e-mail, webmail), how email works (client-server model), File Transfer Protocol – How FTP works (client-server model), Terminal network, Uses of Internet

Unit IV

Programming Languages and Tools, Machine language, Assembly language, High level languages. Problem Solving techniques: Development Tools: Algorithm,

Flowcharts and Pseudo code (Definition and its characteristics) Developing Algorithm and Drawing flowcharts.

C character set, Tokens, Identifiers, Keywords, Variables, Data types, Qualifiers. Operators and Expressions: Arithmetic, Relational, Logical, Bit-Wise, Increment, Decrement, Conditional and Special operators. typedef, Type Conversion, Constants, Declaring Symbolic Constants, Character Strings, Enumerated Data Types, Operator Precedence and Associativity. Library functions: Maths, character and string handling Functions.

Unit V

Control Structure: Compound statement, Selection statement: if, if-else, nested if, switch. Iteration statement: for, while, do..while, Nested loops, Jump statement: break, continue, goto. (Special emphasis on problem solving).

Arrays: Need, Types: Single and Two Dimensional Array. Strings: Strings Manipulation, Arrays of Strings, Evaluation order

Function: Function Components, Return Data type, Parameter Passing, Return by Reference, Default Arguments, Recursive Functions, Arrays with Functions. Macro: single-line and multi-line. Storage Classes.
(Special emphasis on problem solving)

Unit VI

Structure: Declaration, Definition, Accessing structure members, Initialization, Nesting of Structures. Union: Unions, Differences between Structure and Union

Pointer: Introduction, Address Operator (&), Pointer variables, void pointers, Pointer Arithmetic, Pointers to Pointers.

File handling: Hierarchy of File Stream Classes, Opening & closing a file, Testing for errors, File Modes, File pointers and their manipulations, Sequential Access, Random Access, Command Line arguments.

Graphics: Initializing graphics, Drawing basic shapes, Graphics Color system, Types of pens and brushes.

B.Sc. Semester I
COMPUTER SCIENCE PRACTICALS
Course Code: UCS110P

Section A

Minimum five study experiments based on unit I through unit III

Section B

1. Program to compute Fibonacci series.
2. Program to find if a given number is prime or not.
3. Program to accept number and display it in words.
4. Program to find sum of digits of any entered no.
5. Program to reverse the digit.
6. Program to find frequency of occurrence of a given number from an array of N elements.
7. Program to reverse an array.
8. Program to Insert an element in one dimensional array at a given position.
9. Program to Delete an element from one dimensional array.
10. Program to Arrange string data (name of students) in alphabetical order using bubble sort.
11. Program to search the element in an array of N elements using Linear search method & Binary search method.
12. Program to a) Multiply two dimensional Array's (3X3 matrix)
    b) Find largest element in two dimensional Array (3X3matrix)
13. Program to a) Check if given string is Palindrome or not
    b) Calculate number of blanks, vowels and words
14. Program to compute:
    a) Cosine series: $\cos x = 1 - x^2/2! + x^4/4! - x^6/6! +$----------
    b) Sine series: $\sin x = x - x^3/3! + x^5/5! - x^7/7! +$ ---------
15. Program to find factorial of a number using recursive function.
16. Program using function to find sum of two numbers
    a) With no argument & no return values,
    b) With argument & no return values,
    c) With argument & return values.
17. Program to demonstrate passing structure to functions by using
    a) Call by Value and
    b) Call by reference
    Fields names are empno, name, and basic.
18. Program to swap values of two variables by passing pointers.
19. Read two integers and determine bigger of the two with the help offunction

big() returning an integer pointer.

20. Create a sequential file and perform following operation:
    a. Add records b. Process & Display output.
    Field names are Roll, Name, M1, M2, and M3.
21. Program to draw basic shapes like lines, rectangles, ellipse, pie etc. using graphics.
22. Drawing concentric circles and rectangles with various colors.
23. Program to use various pens and brushes using graphics.
24. Creating a text file and copy contents to another file and print it.
25. Create a binary file, store n integers. Read the file and divide it into two different files containing odd and even values separately.
26. Create a menu driven program to add, modify, and delete a record an any entity.

REFERENCE BOOKS:

1. The Art of programming through flowcharts & algorithm by Anil B.Chaudhari, Firewall Media, Laxmi Publication, New Publication.
2. Programming in C by E. Balagurusamy.
3. C Programming – Kernighan & Ritchie
4. C Programming – Dr. Vishal M. Lichade Wiley-Dreamtech Publication
5. Let us C – Y. Kanetkar.
6. C Programming – Holzner, PHI Publication.
7. Programming in C – Ravichandran.
8. Information Technology Concepts by Dr. Madhulika Jain, Shashank & Satish Jain, [BPB Publication, New Delhi.]
9. Fundamentals of Information Technology By Alexis And Mathews Leon [Leon Press, Chennai &Vikas Publishing House Pvt. Ltd, New Delhi]

Web Resources:

Students are advised to make use of the resources available on the Internet. Some useful links related to computer science are given below.

1. www.tutorialspoint.com/cprogramming/
2. www.programiz.com/c-programming
3. www.w3schools.com
4. https://ittutorials.net/
5. http://www.tutorialspoint.com/computer_fundamentals/

Shiksha Mandal's
Bajaj College of Science, Wardha
(Formerly known as Jankidevi Bajaj College of Science, Wardha)

Syllabus of

Semester I

Computer Science Minor

(BSc Physics with Computer Science as Minor/ BSc Electronics with Computer Science as Minor/ BSc Mathematics with Computer Science as Minor)

(Approved in BoS meeting held on 30.03.2024 w.e.f. Academic Session 2024-25)

Course Objective:

There are two main objectives to the Computer Science Minor course.

a) The primary goal of the course in Computer Science is to provide students the foundations for the future work and careers in computation-based courses. These foundations support both a successful career path in computing as well as provide appropriate qualifications for further degree work in computation related disciplines. Our course emphasizes development of analytical skills, acquisition of knowledge and understanding of systems, languages and tools required for effective computation-based problem solving.

b) A Computer Science minor will provide you with the knowledge of computer fundamentals, principles of programming, operating systems, and exposure to various programming languages. These skills will prepare graduates to enter technological fields such as systems programming, technical support, research and teaching.

On completion of the Computer Science course, the students:

a) Will have sound knowledge of the Programming languages such as C, C++, and Java. Also they will have knowledge about Information Technology, Operating System, System Analysis and Design and Development.

b) Can participate in off-campus internships, independent study projects, part-time computer employment, lectures by guest speakers, Society of Computing Students programming and on campus employment as lab assistants, lab supervisors and Helpdesk assistants.

c) Be in a position to develop industrial and entrepreneur applications.

Aim of the Course

To understand the basics of computer hardware, software, network and algorithms, flowcharts and programming and also to provide sound knowledge of object-oriented programming concepts and their understanding, development, working process of software.

Syllabus for SEMESTER I
(To be implemented from Academic Session 2024-25)

Learning Objectives	Course Learning Outcomes
Students will try to learn:	After successful completion of the course student will be able to
1. To understand the concepts of fundamentals of a computer system, number systems, generations of computer languages, and other system tools like translator programs.	1. Understand various number systems, generations of computer languages and translators.
2. To understand various types of memories, storage devices, input and output devices.	2. Know various types of memories, storage devices, input and output devices.
3. To get acquainted with concepts of networks and its associated terminologies, topologies, architecture, communication devices and recent technologies like Bluetooth and Wi-Fi.	3. Learn the concepts of networks, topologies, architecture, communication devices and recent technologies.
4. To understand the basics of 'C' programming. Also get knowledge of algorithm and flowcharts.	4. Learn to make basics programs and also able to draw algorithm and flowcharts of programs.
5. To know the concepts of control and looping structure. Also, student get the idea about concepts of arrays and functions.	5. Perform programs using control and looping structure also understand the basic programs on arrays and functions.
6. To understand the basics of structures, pointers and file handling.	6. Implementation of structures, pointers and file handling concepts in programming.

Minor - I
FUNDAMENTALS OF INFORMATION TECHNOLOGY and PROGRAMMING IN 'C'
Course Code: UCS111T

Credits: 6 (4 Th, 2 Pr) **No. of Lectures: 60** **No. of Practical: 60**

Course Outline

Unit I

Evolution of Computers, Generations of Computer, Characteristics of a Computer, Computer Classification.

Basic Components of Digital Computers: Block Diagram. CPU: Functions of Each Unit: Primary Memory, ALU and CU, Instruction format. Bus: Data, Control and Address Bus. Number Systems: Binary, Octal, Decimal, Hexadecimal, Their Conversions, Binary Arithmetic. ASCII, BCD, EBCDIC.

Language Evolution: Generation of Languages: Machine, Assembly, High Level Languages. Characteristics of Good Language Translators: Compiler, Interpreter and Assembler. Source and object Program.

Unit II

Memory: Memory hierarchy, RAM, Static & dynamic RAM, Types of RAM (SDRAM, RDRAM, DDR), ROM, PROM, EPROM, EEPROM, Flash and Cache. Storage Devices: Hard Disk, Zip Disk and Optical Disk. Pen Drive, Blu Ray and SSD.

Input Devices: Keyboard, Mouse, Light Pen, Touch Screen, Voice Input, MICR, OCR, OMR, Barcode Reader and Flatbed Scanner. Output Devices: VDU, Printers: Dot Matrix, Laser and Inkjet. Plotters: Drum, Flat-Bed and Inkjet.

Unit III

Computer Network: Goals and applications - Business Application, Home Application, Network terminology, Topologies: Linear, Circular, Tree and Mesh. Types of Networks: LAN, WAN, MAN. Repeaters, Bridge, Routers, Brouters and Gateway. Modem for Communication between PC's, Wi-Fi network, Introduction of Bluetooth and Infrared devices. Network protocols. Architecture: Peer-to-Peer, Client/Server. TCP/IP Reference Model with functionality of each layer.

Internet services (Introduction only): WWW – Web browser, URL, Internet search engines, WWW development languages, Electronic mail – E-mail address, e-mail message format, e-mail services (application based e-mail, webmail), how email works (client-server model), File Transfer Protocol – How FTP works (client-server model), Terminal network, Uses of Internet

Unit IV

Programming Languages and Tools, Machine language, Assembly language, High level languages. Problem Solving techniques: Development Tools: Algorithm, Flowcharts and Pseudo code (Definition and its characteristics) Developing Algorithm and Drawing flowcharts.

C character set, Tokens, Identifiers, Keywords, Variables, Data types, Qualifiers. Operators and Expressions: Arithmetic, Relational, Logical, Bit-Wise, Increment, Decrement, Conditional and Special operators. typedef, Type Conversion, Constants, Declaring Symbolic Constants, Character Strings, Enumerated Data Types, Operator Precedence and Associativity. Library functions: Maths, character and string handling Functions.

Unit V

Control Structure: Compound statement, Selection statement: if, if-else, nested if, switch. Iteration statement: for, while, do..while, Nested loops, Jump statement: break, continue, goto. (Special emphasis on problem solving).

Arrays: Need, Types: Single and Two Dimensional Array. Strings: Strings Manipulation, Arrays of Strings, Evaluation order

Function: Function Components, Return Data type, Parameter Passing, Return by Reference, Default Arguments, Recursive Functions, Arrays with Functions. Macro: single-line and multi-line. Storage Classes.
(Special emphasis on problem solving)

Unit VI

Structure: Declaration, Definition, Accessing structure members, Initialization, Nesting of Structures. Union: Unions, Differences between Structure and Union

Pointer: Introduction, Address Operator (&), Pointer variables, void pointers, Pointer Arithmetic, Pointers to Pointers.

File handling: Hierarchy of File Stream Classes, Opening & closing a file, Testing for errors, File Modes, File pointers and their manipulations, Sequential Access, Random Access, Command Line arguments.

Graphics: Initializing graphics, Drawing basic shapes, Graphics Color system, Types of pens and brushes.

Semester I
COMPUTER SCIENCE PRACTICAL
Course Code: UCS111P

Section A

Minimum five study experiments based on unit I through unit III.

Section B

1. Program to compute Fibonacci series.
2. Program to find if a given number is prime or not.
3. Program to accept number and display it in words.
4. Program to find sum of digits of any entered no.
5. Program to reverse the digit.
6. Program to find frequency of occurrence of a given number from an array of N elements.
7. Program to reverse an array.
8. Program to Insert an element in one dimensional array at a given position.
9. Program to Delete an element from one dimensional array.
10. Program to Arrange string data (name of students) in alphabetical order using bubble sort.
11. Program to search the element in an array of N elements using Linear search method & Binary search method.
12. Program to a) Multiply two dimensional Array's (3X3 matrix)
    b) Find largest element in two-dimensional Array (3X3matrix)
13. Program to a) Check if given string is Palindrome or not
    b) Calculate number of blanks, vowels and words
14. Program to compute:
    a) Cosine series: $\cos x = 1 - x^2/2! + x^4/4! - x^6/6! +$----------
    b) Sine series: $\sin x = x - x^3/3! + x^5/5! - x^7/7! +$ ---------
15. Program to find factorial of a number using recursive function.
16. Program using function to find sum of two numbers
    a) With no argument & no return values,
    b) With argument & no return values,
    c) With argument & return values.
17. Program to demonstrate passing structure to functions by using
    a) Call by Value and
    b) Call by reference
    Fields names are empno, name, and basic.
18. Program to swap values of two variables by passing pointers.

19. Read two integers and determine bigger of the two with the help of function big() returning an integer pointer.
20. Create a sequential file and perform following operation:
    a. Add records b. Process & Display output.
    Field names are Roll, Name, M1, M2, and M3.
21. Program to draw basic shapes like lines, rectangles, ellipse, pie etc. using graphics.
22. Drawing concentric circles and rectangles with various colors.
23. Program to use various pens and brushes using graphics.
24. Creating a text file and copy contents to another file and print it.
25. Create a binary file, store n integers. Read the file and divide it into two different files containing odd and even values separately.
26. Create a menu driven program to add, modify, and delete a record an any entity.

REFERENCE BOOKS:

1. The Art of programming through flowcharts & algorithm by Anil B.Chaudhari, Firewall Media, Laxmi Publication, New Publication.
2. Programming in C by E. Balagurusamy.
3. C Programming – Kernighan & Ritchie
4. C Programming – Dr. Vishal M. Lichade Wiley-Dreamtech Publication
5. Let us C – Y. Kanetkar.
6. C Programming – Holzner, PHI Publication.
7. Programming in C – Ravichandran.
8. Information Technology Concepts by Dr. Madhulika Jain, Shashank & Satish Jain, [BPB Publication, New Delhi].
9. Fundamentals of Information Technology By Alexis And Mathews Leon [Leon Press, Chennai &Vikas Publishing House Pvt. Ltd, New Delhi].

Web Resources:

Students are advised to make use of the resources available on the Internet. Some useful links related to computer science are given below.

1. www.tutorialspoint.com/cprogramming/
2. www.programiz.com/c-programming
3. www.w3schools.com
4. https://ittutorials.net/
5. http://www.tutorialspoint.com/computer_fundamentals/

Shiksha Mandal's
Bajaj College of Science, Wardha
(Formerly known as Jankidevi Bajaj College of Science, Wardha)

Syllabus of

Bachelor of Science (B. Sc.) Three Year (Six Semesters) Degree Course

Semester II

Subject: Computer Science Major

(Approved in BoS meeting held on 15.05.2023 w.e.f. Academic Session2023-24)

Programme Objective:

There are two main objectives to the B.Sc. Computer Science Programme.

a) The primary goal of the degree program in Computer Science is to provide students the foundations for the future work and careers in computation-based courses. These foundations support both a successful career path in computing as well as provide appropriate qualifications for further degree work in computation related disciplines. Our degree emphasizes development of analytical skills, acquisition of knowledge and understanding of systems, languages and tools required for effective computation-based problem solving.

b) A Computer Science major will provide you with knowledge of programming, operating systems, compiler design and principles of programming language. These skills will prepare graduates to enter technological fields such as systems programming, technical support, research and teaching.

On completion of the B.Sc. Computer Science Programme, the students:

a) Will have sound knowledge of the Programming languages such as C, C++, Java, VB and SQL. Also they will have knowledge about Information Technology, Operating System, System Analysis and Design and Development.

b) Can participate in off-campus internships, independent study projects, part-time computer employment, lectures by guest speakers, Society of Computing Students programming and on campus employment as lab assistants, lab supervisors and Helpdesk assistants.

c) Be in a position to develop industrial and entrepreneur applications.

Aim of the Course

To understand the basics of computer hardware, software, network and algorithms, flowcharts and programming and also to provide sound knowledge of object oriented programming concepts and their understanding, development, working process of software.

Syllabus for B.Sc. SEMESTER II
(To be implemented from Academic Session 2023-24)

Learning Objectives	Course Learning Outcomes
Students will try to learn:	After successful completion of the course student will be able to
1. To introduce the basics of Object Oriented Programming like classes, objects, abstraction, encapsulation and polymorphism, etc.	1. Learn to write programs using Object Oriented Programming concepts like class and object.
2. To understand the concepts of constructor and destructor; also familiar with operator overloading and dynamic objects.	2. Implementation of creation, deletion of objects and to make dynamic objects.
3. To get an idea of inheritance, virtual functions and exception handling.	3. Understand the implementation of inheritance, virtual functions and exception handling using programming.
4. To know about components of Computerized Information Systems, role of Systems Administrator, various software development models; structured tools and techniques used for data collection and data analysis.	4. Understand components of Computerized Information Systems, role of Systems Administrator, various software development models; structured tools and techniques.
5. To understand concepts associated with input and output design, form design, activities related to implementation and change strategies related to a computer system.	5. Know concepts associated with input and output design, form design, implementation activities and change strategies.
6. To understand the principles of testing strategies, conversion methods, planning and scheduling techniques, SCM, ISO, software re-engineering and software maintenance costs.	6. Get acquainted with the principles of testing strategies, conversion methods, planning and scheduling techniques, and advanced concepts.

DSC-II

OBJECT ORIENTED PROGRAMMING USING 'C++' and SYSTEM ANALYSIS AND DESIGN

Course Code: UCS120T

Credits: 6 (4 Th, 2 Pr) **No. of Lectures: 60** **No. of Practical: 60**

Course Outline

Unit I

Object Oriented Methodology: Elements of Object Oriented programming, Objects, Classes, OOPs features.

Classes & Objects: Specifying a Class, Creating Objects, Accessing Class members, Defining member function, Outside Member Functions as inline, Accessing Member Functions within the class, Static data member, Access Specifiers: Private, Protected and Public Members.

Unit II

Constructors & Destructors: Introduction, Parameterized Constructors, Constructor Overloading, Constructors with Default Arguments, Copy Constructor, Dynamic Constructor, Destructor, Order of Construction and Destruction, Static data members with Constructor and Destructors, Virtual Destructor.

Operator Overloading: Definition, Overloadable Operators, Unary Operator Overloading, Binary overloading, Rules for Operator Overloading.

Dynamic Objects: Pointers to Objects, Creating and Deleting Dynamic Objects: New and Delete operators, Array of Objects, Array of Pointers to Objects, Pointers to Object Members, this Pointer.

Unit III

Inheritance: Definition, Abstract classes, Single, Multilevel, Multiple, Hierarchical, Hybrid Inheritance, Constructor and Destructor in Derived Classes.

Virtual Functions: Need for Virtual Functions, definition, Pure Virtual Functions, Abstract Classes, Rules for Virtual Function. Runtime polymorphism.

Exception Handling: Exception Handling Model, List of Exceptions, catch with ellipses catch( ) Handling Uncaught Exceptions, Fault Tolerant Design Techniques, Memory Allocation Failure Exception, Rules for Handling Exception Successfully.

Unit IV

Introduction: System, Subsystems, Components of Computerized Information System, Systems Analysts, SDLC, Prototyping model and Spiral model.

Feasibility Study and Analysis: Identifying Problems, Organizing Feasibility Analysis: Economic, Financial, Organizational and Technological. Feasibility Decision, Choice of a solution.

Data Collection: Interviews, Brain Storming, Questionnaires, Document Search, Observation.

Structured tools and techniques of Data analysis: Structured English, Process Charts, SOP, Decision Tables and Decision Trees, Data Flow Diagram, Data Dictionary (Special emphasis on problem solving).

Unit V

System Design: Input design: Input Validation, Human factor Consideration, Messages, System Tolerance.

Output design: Categories of output, Design Principles, Control of Output. Forms: Principles of Form Design, Ways to ensure Quality Forms.

Codes: Types, Physical Representation of Codes, Principle of Code Design. Implementation: Training, Operational Training and Related Activities,
Planning to Implement Change, Change Strategies.

Unit VI

Testing: Preparation for Testing, Test Execution: Levels of Testing, Component, Function, Subsystem, System, Test Evaluation, Acceptance.

System Evaluation Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Scheduling: Work Breakdown Structure, Activity Networks and CPM, Gantt Charts, PERT Charts, Project Monitoring and Control. Risk Management.

Software Configuration Management: Necessity, Configuring Management Activities Software Reliability and Quality Management: Software Reliability, Software Quality, ISO 9000. Software Maintenance: Characteristics of Software Maintenance, Maintenance Process Models, Estimation of Maintenance Cost. Software Reuse: What can be reused, Why no reuse so far, Basic Issues.

B. Sc. Semester II
COMPUTER SCIENCE PRACTICALS
Course Code: UCS120P

Section A

1. Demonstrate the usage of Constructor and Destructor. Define a class data with data member acct_no, balance containing constructor data to initialize data member and a member function display () for output.

2. Program to demonstrate usage of a constructor and Destructor function. Declare a class with public data member count. The class containing one constructor and destructor to maintain updated information about active objects i.e. i) No of objects created. ii) No of objects Destroyed.

3. Program to accept the distance between city 1st & 2nd, city 2nd & 3rd. Calculate the distance between city 1st & 3rd. Define a class road with private data member km, m, d1, d2, d3 containing member function getdata () to accept values of d1, d2 and calculate for calculating distance.

4. Demonstrate the use of operators overloading (string manipulation: + for concatenation and relational operators for alphabetical comparison).

5. In a bank N depositor deposit the amount. Write a program to find total amount deposited in the bank. Declare a class deposit with private data member Rupee and Paisa containing member function getdata(), putdata().Use array of objects, use operator '+' for overloading.

6. Declare class event and accept time of first event and second event and find the difference between 1st and 2nd event containing public member function getdata() and display with private data member hour, minute, second and total. Use operator '-'for overloading.

7. Program to demonstrate Single Inheritance. Declare a class B and derive publically class D from B.
    a. The class B contains private data member a, public data member b with member function get_ab(), get_a(), show_a().
    b. The derived class D contains data member c with member function mul()and display().

8. Program to demonstrate Multiple Inheritances. Declare class M and N and derive publically class P from M and N.
    a. Declare a class M with protected data member m and public member function get_m().

b. Declare a class N with protected data member n containing member function get_n().
c. Declare class P containing member function display ().

9. Program to demonstrate Multilevel Inheritance. Declare a class student and derive publically a class test and derive publically class result from class test.
    a. The class student contains protected data member roll_number with public member functions get_number() and put_number().

    b. The class test containing protected data member sub1, sub2 with public member function get_marks() and put_marks().
    c. The class result contains data member total and public member function display ().

10. Program to demonstrate Hierarchical Inheritance. Declare a class Side and derive publically class Square from base class Side and also derive publically class Cube from base class Side.
    a. Class Side contains protected data member L with a member function set_values().
    b. Class Square contains member function sq().
    c. Class Cube contains member function cub().

11. Program to demonstrate usage of normal virtual function and pure virtual function with abstract class.

12. Program to determine whether the input is +ve or –ve through exception.

13. Program to raise exception if an attempt is made to perform divide by zero.
14. Program to accept customer account information like Customer Name, Account Number, Account Type. Implement the following operations using inheritance:
    a. Deposit
    b. Withdraw
    c. Display Balance
    d. Display Account details.

Section B

Minimum five study experiments based on unit IV through Unit VI

REFERENCE BOOKS:

1. Mastering C++ by K R Venugopal Tata McGraw-Hill, New Delhi.
2. The C++ Programming Language –Bjarne Stroustrup
3. Programming with C++ - Ravichandran
4. Programming with C++ - Robert Lafore
5. Object Oriented Programming with C++ by E. Balagurusamy, McGraw- Hill
6. Information Systems Analysis, Design and Implementation By K. M. Hussain, Donna Hussain [Tata McGraw-Hill Publishing Company Ltd, New Delhi]

7. Fundamentals of Software Engineering by Rajib Mall [PHI Publication]
8. Workbook on Systems Analysis & Design by V. Garg [PHI Publication]
9. System Analysis and Design- Don Yeates, Shiebls, Helmy (M).
10. System Analysis & Design - Edward –TMH
11. System Analysis and Design – Satzinger, Robert Jackson and Stephen Burd, Thomson Learning

Web Resources:

Students are advised to make use of the resources available on the Internet. Some useful links related to computer science are given below.

1. www.tutorialspoint.com/cplusplus
2. www.cplusplus.com/doc/tutorial
3. www.cprogramming.com/tutorial
4. www.subjectcoach.com/tutorials
5. www.tutorialspoint.com/sdlc

Shiksha Mandal's
Bajaj College of Science, Wardha
(Formerly known as Jankidevi Bajaj College of Science, Wardha)

Syllabus of

Semester II

Computer Science Minor

(BSc Physics with Computer Science as Minor/ BSc Electronics with Computer Science as Minor/ BSc Mathematics with Computer Science as Minor)

(Approved in BoS meeting held on 30.03.2024 w.e.f. Academic Session 2024-25)

Course Objective:

There are two main objectives to the Computer Science Minor course.

a) The primary goal of the course in Computer Science is to provide students the foundations for the future work and careers in computation-based courses. These foundations support both a successful career path in computing as well as provide appropriate qualifications for further degree work in computation related disciplines. Our course emphasizes development of analytical skills, acquisition of knowledge and understanding of systems, languages and tools required for effective computation-based problem solving.

b) A Computer Science minor will provide you with the knowledge of computer fundamentals, principles of programming, operating systems, and exposure to various programming languages. These skills will prepare graduates to enter technological fields such as systems programming, technical support, research and teaching.

On completion of the Computer Science course, the students:

a) Will have sound knowledge of the Programming languages such as C, C++, and Java. Also, they will have knowledge about Information Technology, Operating System, System Analysis and Design and Development.

b) Can participate in off-campus internships, independent study projects, part-time computer employment, lectures by guest speakers, Society of Computing Students programming and on campus employment as lab assistants, lab supervisors and Helpdesk assistants.

c) Be in a position to develop industrial and entrepreneur applications.

Aim of the Course

To understand the basic and advanced concepts of object-oriented programming and their implementation using C++, Systems concept, various software development models, tools and techniques used for software design.

Syllabus for SEMESTER II
(To be implemented from Academic Session 2024-25)

Learning Objectives	Course Learning Outcomes
Students will try to learn:	After successful completion of the course student will be able to
1. To introduce the basics of Object Oriented Programming like classes, objects, abstraction, encapsulation and polymorphism, etc.	1. Learn to write programs using Object Oriented Programming concepts like class and object.
2. To understand the concepts of constructor and destructor; also familiar with operator overloading and dynamic objects.	2. Implementation of creation, deletion of objects and to make dynamic objects.
3. To get an idea of inheritance, virtual functions and exception handling.	3. Understand the implementation of inheritance, virtual functions and exception handling using programming.
4. To know about components of Computerized Information Systems, role of Systems Administrator, various software development models; structured tools and techniques used for data collection and data analysis.	4. Understand components of Computerized Information Systems, role of Systems Administrator, various software development models; structured tools and techniques.
5. To understand concepts associated with input and output design, form design, activities related to implementation and change strategies related to a computer system.	5. Know concepts associated with input and output design, form design, implementation activities and change strategies.
6. To understand the principles of testing strategies, conversion methods, planning and scheduling techniques, SCM, ISO, software re-engineering and software maintenance costs.	6. Get acquainted with the principles of testing strategies, conversion methods, planning and scheduling techniques, and advanced concepts.

Minor-II

OBJECT ORIENTED PROGRAMMING USING 'C++' and SYSTEM ANALYSIS AND DESIGN

Course Code: UCS121T

Credits: 6 (4 Th, 2 Pr) **No. of Lectures: 60** **No. of Practical: 60**

Course Outline

Unit I

Object Oriented Methodology: Elements of Object Oriented programming, Objects, Classes, OOPs features.

Classes & Objects: Specifying a Class, Creating Objects, Accessing Class members, Defining member function, Outside Member Functions as inline, Accessing Member Functions within the class, Static data member, Access Specifiers: Private, Protected and Public Members.

Unit II

Constructors & Destructors: Introduction, Parameterized Constructors, Constructor Overloading, Constructors with Default Arguments, Copy Constructor, Dynamic Constructor, Destructor, Order of Construction and Destruction, Static data members with Constructor and Destructors, Virtual Destructor.

Operator Overloading: Definition, Overloadable Operators, Unary Operator Overloading, Binary overloading, Rules for Operator Overloading.

Dynamic Objects: Pointers to Objects, Creating and Deleting Dynamic Objects: New and Delete operators, Array of Objects, Array of Pointers to Objects, Pointers to Object Members, this Pointer.

Unit III

Inheritance: Definition, Abstract classes, Single, Multilevel, Multiple, Hierarchical, Hybrid Inheritance, Constructor and Destructor in Derived Classes.

Virtual Functions: Need for Virtual Functions, definition, Pure Virtual Functions, Abstract Classes, Rules for Virtual Function. Runtime polymorphism.

Exception Handling: Exception Handling Model, List of Exceptions, catch with ellipses catch() Handling Uncaught Exceptions, Fault Tolerant Design Techniques, Memory Allocation Failure Exception, Rules for Handling Exception Successfully.

Unit IV

Introduction: System, Subsystems, Components of Computerized Information System, Systems Analysts, SDLC, Prototyping model and Spiral model.

Feasibility Study and Analysis: Identifying Problems, Organizing Feasibility Analysis: Economic, Financial, Organizational and Technological. Feasibility Decision, Choice of a solution.

Data Collection: Interviews, Brain Storming, Questionnaires, Document Search, Observation.

Structured tools and techniques of Data analysis: Structured English, Process Charts, SOP, Decision Tables and Decision Trees, Data Flow Diagram, Data Dictionary (Special emphasis on problem solving).

Unit V

System Design: Input design: Input Validation, Human factor Consideration, Messages, System Tolerance.

Output design: Categories of output, Design Principles, Control of Output. Forms: Principles of Form Design, Ways to ensure Quality Forms.

Codes: Types, Physical Representation of Codes, Principle of Code Design. Implementation: Training, Operational Training and Related Activities,
Planning to Implement Change, Change Strategies.

Unit VI

Testing: Preparation for Testing, Test Execution: Levels of Testing, Component, Function, Subsystem, System, Test Evaluation, Acceptance.

System Evaluation Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Scheduling: Work Breakdown Structure, Activity Networks and CPM, Gantt Charts, PERT Charts, Project Monitoring and Control. Risk Management.

Software Configuration Management: Necessity, Configuring Management Activities Software Reliability and Quality Management: Software Reliability, Software Quality, ISO 9000. Software Maintenance: Characteristics of Software Maintenance, Maintenance Process Models, Estimation of Maintenance Cost. Software Reuse: What can be reused, Why no reuse so far, Basic Issues.

Semester II
COMPUTER SCIENCE PRACTICAL
Course Code: UCS121P

Section A

1. Demonstrate the usage of Constructor and Destructor. Define a class data with data member acct_no, balance containing constructor data to initialize data member and a member function display () for output.

2. Program to demonstrate usage of a constructor and Destructor function. Declare a class with public data member count. The class containing one constructor and destructor to maintain updated information about active objects i.e. i) No of objects created. ii) No of objects Destroyed.

3. Program to accept the distance between city 1st & 2nd, city 2nd & 3rd. Calculate the distance between city 1st & 3rd. Define a class road with private data member km, m, d1, d2, d3 containing member function getdata () to accept values of d1, d2 and calculate for calculating distance.

4. Demonstrate the use of operators overloading (string manipulation: + for concatenation and relational operators for alphabetical comparison).

5. In a bank N depositor deposit the amount. Write a program to find total amount deposited in the bank. Declare a class deposit with private data member Rupee and Paisa containing member function getdata(), putdata().Use array of objects, use operator '+' for overloading.

6. Declare class event and accept time of first event and second event and find the difference between 1st and 2nd event containing public member function getdata() and display with private data member hour, minute, second and total. Use operator '-'for overloading.

7. Program to demonstrate Single Inheritance. Declare a class B and derive publically class D from B.
    a. The class B contains private data member a, public data member b with member function get_ab(), get_a(), show_a().
    b. The derived class D contains data member c with member function mul()and display().

8. Program to demonstrate Multiple Inheritances. Declare class M and N and derive publically class P from M and N.

   a. Declare a class M with protected data member m and public member function get_m().
   b. Declare a class N with protected data member n containing member function get_n().
   c. Declare class P containing member function display ().

9. Program to demonstrate Multilevel Inheritance. Declare a class student and derive publically a class test and derive publically class result from class test.
   a. The class student contains protected data member roll_number with public member functions get_number() and put_number().
   b. The class test containing protected data member sub1, sub2 with public member function get_marks() and put_marks().
   c. The class result contains data member total and public member function display ().

10. Program to demonstrate Hierarchical Inheritance. Declare a class Side and derive publically class Square from base class Side and also derive publically class Cube from base class Side.
   a. Class Side contains protected data member L with a member function set_values().
   b. Class Square contains member function sq().
   c. Class Cube contains member function cub().

11. Program to demonstrate usage of normal virtual function and pure virtual function with abstract class.

12. Program to determine whether the input is +ve or –ve through exception.

13. Program to raise exception if an attempt is made to perform divide by zero.
14. Program to accept customer account information like Customer Name, Account Number, Account Type. Implement the following operations using inheritance:
   a. Deposit
   b. Withdraw
   c. Display Balance
   d. Display Account details.

Section B

Minimum five study experiments based on unit IV through Unit VI

REFERENCE BOOKS:

1. Mastering C++ by K R Venugopal Tata McGraw-Hill, New Delhi.
2. The C++ Programming Language –Bjarne Stroustrup
3. Programming with C++ - Ravichandran
4. Programming with C++ - Robert Lafore
5. Object Oriented Programming with C++ by E. Balagurusamy, McGraw- Hill

6. Information Systems Analysis, Design and Implementation By K. M. Hussain, Donna Hussain [Tata McGraw-Hill Publishing Company Ltd, New Delhi]
7. Fundamentals of Software Engineering by Rajib Mall [PHI Publication]
8. Workbook on Systems Analysis & Design by V. Garg [PHI Publication]
9. System Analysis and Design- Don Yeates, Shiebls, Helmy (M).
10. System Analysis & Design - Edward –TMH
11. System Analysis and Design – Satzinger, Robert Jackson and Stephen Burd, Thomson Learning

Web Resources:

Students are advised to make use of the resources available on the Internet. Some useful links related to computer science are given below.

1. www.tutorialspoint.com/cplusplus
2. www.cplusplus.com/doc/tutorial
3. www.cprogramming.com/tutorial
4. www.subjectcoach.com/tutorials
5. www.tutorialspoint.com/sdlc