BCA 5TH SEMESTER(CBCS) SYLLABUS

Semester	CORE COURSE (14)	Ability Enhancement Compulsory Course (AECC) (2)	Skill Enhancement Course (SEC) (2)	Elective: Discipline Specific (DSE) (4)	Elective: Generic (GE) (4)
V	BCA-HC-5016 Java Programming			DSE-1	
	BCA-HC-5026 Operating System			DSE -2	

Paper Code: CIT-HC-1016: Means: CIT (Subject code), HC (Course type: Honours Core), 1(Semester), 01(first paper of the semester), 6(credit).

Discipline Specific Electives (DSE)

DSE-1

(i)BCA-HE-5016: Project Work / Dissertation (Credit: 6)

DSE-2 (choose any One)

(i) BCA-HE-5026:Data Mining & Warehousing
(ii) BCA-HE-5036: Computer Oriented Numerical Methods and statistical Techniques
(iii) BCA-HE-5046: Programming in Python

BCA-HC-5016: JAVA PROGRAMMING

(Credit: 4+2=6)(L: 4, P: 4, T: 0)

Theory: 60 Lectures, Practical: 60 Lectures

UNIT 1: JAVA language basics **(12 Lectures)**

Basic features, Java virtual machine concepts Creation of JAVA, executing a java program using command line arguments, The primitive data types and Variables, Java Key words, integer and floating point data type, character and Boolean types, declaring and initialization variables, Type conversion and casting

UNIT 2: Operators and Control Statements **(12 Lectures)**

Java operators - Arithmetic operators, Bitwise operators, Relational operators, Boolean logical operators, Assignment operator, Conditional operator, if and switch statements, iteration statements, jump statements.

UNIT 3: Classes and Methods **(15 Lectures)**

Class fundamentals, Objects, Constructors, this keyword, finalize () method, Overloading methods, garbage collection, Returning objects, introducing access control, understanding static, introducing final, introducing nested and inner classes, String operations, Character Extraction, Comparing, Searching & Modifying the strings, Data conversion using valueOf(), StringBuffer

UNIT 4: Inheritance **(12 Lectures)**

Inheritance basics, using super, creating a multilevel hierarchy, method overriding, dynamic method dispatch, using abstract classes, using final with inheritance Packages and interfaces Packages, access protection, importing packages, interfaces Multithread programming, The JAVA thread model, creating a thread, creating a multiple thread, Using is Alive() and join (), Inter thread communication, suspending, resuming and stopping threads, using multithreading.

UNIT 5: Exception handling **(12 Lectures)**

Exception handling fundamentals, exception types, uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, throw, throws, finally, Java's built-in Exceptions, Input/output: Java I/O classes and interfaces, file, the stream classes, byte streams, character streams, console class. Applet class: Applet basics, applet architecture, simple applet skeleton, applet displaying methods, Event handling: Two event handling mechanisms, delegation event model, event classes, source of events, event listener interface

Practical / Lab work to be performed

Each student should do at least 10 assignments from the following list.

1. Design a class to represent a bank account and include the following data members –

Data Members: name of the depositor, account number, type of a/c, balance amount in the a/c

Methods: to assign initial values, to deposit an amount, to withdraw an amount after checking the minimum balance (Rs.1000), to display the name of the depositor and balance.

2. Write an applet programming to print the first name, last name, sex, address, mobile no. and pin code of an end user passing parameters.

3. Write an applet programming to create three buttons and draw a rectangle on clicking the first button, a solid rounded rectangle on clicking the second button and a solid circle and an arc on clicking the third button.

4. Write an applet program to draw the following shapes –

a) A straight line b) A polygon d) A solid oval e) A solid rounded rectangle f) A polyline

5. Write a program to create 3 – threads for execution with different priorities.

6. Write a program to create three threads for execution of the natural nos. less than 5 using synchronization concept.

7. Write a program to – a) Print the name of the thread, and its priority

b) Change the name of the current thread to "JAVA"

c) Display the detail of the current thread

8. Write a java program for a class teacher that contains two fields name and qualification. Extend the class to department that contains data members deptno and deptname. An interface name as college contains one field name of the college. Using the above classes and interface get the appropriate information and display them.

9. Design three classes person, employee and student using the concept of inheritance. Each class should have a constructor of its own properties as name, age, gender and common method showdata().

10. Write a program to create a class shape with properties length and breadth. Extend the class to rectangle and square, and find the area of the rectangle and the square. Use input() method to take input using keyboard.

11. Write a program to create an array of employee name and salary related to the employee. If the salary is greater than Rs.10,000 raise an exception "Salary is greater than Rs.10,000", otherwise display the required information.

12. Write a program to find the square root of a number. If the input value is negative, raise a user defined exception "The number is a negative number".

13. Write a program to create three StringBuffers. The first one takes no parameters, second one takes an integer value and the third one sets an initial value "Java". Find the content, length and capacity for the StringBuffers.

14. Write a java program to input a string and converts the string to lower case and upper case. Also find the substring from 5 to end, from 0 to 5, from 3 to 7, and from 5 to 5.

15. Write a java program to create a class, library that contains the field, bookno, Extend the class, library to book having fields author and title, and then extend the class, book to issue. Create an interface, language that contains a field, lang. Implement the interface for the class, book. Use appropriate methods for the classes and interface.

REFRERENCE BOOKS

1. Herbert Schildt, *The Complete Reference*, Seventh Edition, Tata McGraw Hill, 2007.
2. Bruce, Eckel, *Thinking in Java*, Third edition, Pearson Education, 2005

BCA-HC-5026: OPERATING SYSTEM

(Credit: 4+2=6)(L: 4, P: 4, T: 0)

Theory: 60 Lectures, Practical: 60 Lectures

UNIT 1: Introduction **(6 Lectures)**

Basics of Operating Systems: Definition – Generations of operating systems, Types of Operating Systems (definition only): Mainframe, Batch, Multiprocessor, Distributed, Multitasking, Real time, Parallel and Time sharing.

UNIT 2: Processes **(6 Lectures)**

Process: Concept of a Process, Process States, Process creation, Process termination, Context switching, Thread: Concept of thread, Design issues of thread, Types of threads, Benefits of threads, Basic Concept of multithreading.

UNIT 3: Process Synchronization **(6 Lectures)**

Basic concept of Inter-Process communication, Race condition, Critical-Section, Mutual exclusion, semaphore, Mutex, Different ways to achieve mutual exclusion- Disabling interrupt, Test-and-Set Lock, Peterson's solution using semaphore, Brief discussion on classical IPC problem (example Dinning philosopher problem).

UNIT 4: Scheduling **(6 Lectures)**

Basic Concepts of scheduling, Scheduling objectives, pre-emptive and non pre-emptive scheduling, Scheduling criteria – CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time, Basic concepts on batch, interactive and real-time scheduling algorithm, Scheduling algorithms-FCFS, SJF, RR, priority scheduling, Goals of scheduling algorithms.

UNIT 5: Deadlocks **(6 Lectures)**

Definition, Deadlock characteristics , Methods for Handling Deadlocks, Deadlock Prevention, Deadlock detection and Recovery, Deadlock Avoidance using Banker’s Algorithm.

UNIT 6: Memory management **(6 Lectures)**

Memory allocation in Multiprogramming, Relocation and Protection, Swapping, Virtual memory: Basics of Virtual Memory, Logical versus Physical address space, Paging and Concept of Segmentation, Page fault, Page table and its entries, Demand paging, TLB, Page replacement algorithms: LRU, Optimal, NRU, FIFO, Second chance, Clock, NFU, Working set.

UNIT 7: File system **(6 Lectures)**

File concepts, File naming, File types(directory, regular, device), File attributes, Operations on file, Access Methods – Sequential, Random access, Directory in UNIX, Hierarchical directory structure, Relative path and Absolute path, Operation on directories, Disk layout, Disk partition, File system layout, Disk block allocation- Contiguous allocation, Linked list allocation, FAT, i-nodes, File system security

UNIT 8: I/O management **(6 Lectures)**

Basic principles and overall structure of I/O management subsystem, Device controllers, Layers of the I/O subsystem-interrupt handler's device driver, device independent I/O software and user space I/O software.

Practical / Lab work to be performed

Each student should do at least 12 assignments from the following list.

1. Write a program to create a child process that starts looping and then terminates.
2. Write a program to show that the child can be set up to ignore a signal from its parent.
3. Write a program to show that a process can ignore a signal.
4. Write a program to create a thread in which prints "We are proud to be Indians" and terminates.
5. Write a program to demonstrate how to "wait" for thread completions by using the P thread join routine. Threads are explicitly created in a joinable state.
6. Write a program to create a thread in which print "We are proud to be Indians" and pass multiple arguments using structure during its creation.
7. Write a program to compute the dot product of two vectors and also show the use of mutex variable.
8. Write a program to create threads, the main thread creates three threads. Two of these threads increment a counter variable while third thread watches the value of the counter variable. When the counter variable reaches a predefined limit, the waiting thread is signalled by one of the incrementing threads. The waiting thread "awakens" and then modifies the counter. The program continues until the incrementing threads reach a final value and also print the final value.
9. Write a program to show attaching and detaching shared memory.
10. Write a program to show the communication between two processes through shared memory.
11. Write a program to show how two processes can talk to each other using wait() and signal() operations applied on semaphore.
12. Write a program in which a parent process accepts a list of integers to be sorted. Parent process uses the fork system call to create a new process called a child process. Both the processes use shared memory for the list of integers. Now use the parent process to sort the integers using bubble sort and the child process to sort the integers using selection sort. Use semaphore variable for process synchronization.
13. Write a program to implement Banker's Algorithm for multiple resource type each.
14. Write a program to simulate Dining Philosophers Algorithm

REFERENCE BOOKS

1. Tannenbaum, *Operating Systems*, PHI, 4th Edition, 2000
2. Silbersachatz, Galvin, *Operating System Concepts*, Person, 5th Edition, 2001
3. William Stallings, *Operating System*, Prentice Hall of India, 4th Edition, 2003

BCA-HE-5016: PROJECT WORK/DEISSERTATION (Credit: 6)

The students will be allowed to work on any project based on the concepts studied in core / elective or skill based elective courses. The objective of the project is to train the student to independently search, identify and study real-life important topics in CS/IT; to develop skills among students in a particular field of CS/IT; and to expose students to the world of technology, innovation, and research. The problem should be such that the students get a chance to explore one or two technologies in depth and grab good command over those technologies after successful completion of the project. Application problems, if found interesting and arisen at the demand of a particular situation, may also be assigned; but typical information management systems with just two or three simple database tables and/or data- entry forms are to be discouraged.

The group size should be maximum three (03) students. Each group will be assigned a teacher as a supervisor who will handle both their theory as well lab classes. The work will have to be submitted in the form of a dissertation.

A maximum of Four (04) projects would be assigned to one teacher.

BCA-HE-5026: DATA MINING AND WAREHOUSING

(Credit: 5+1=6)(L: 5, P: 0, T: 1)

Theory: 60 Lectures, Tutorial: 15 Lectures

UNIT 1: Introduction to Data Warehousing (12 Lectures)

Need for Data Warehousing, Basic elements of Data Warehousing, differences between Database Systems and Data Warehouse. Data Warehouse Architecture and its components, Infrastructure and metadata. Data Design and Data Representation - Principles of dimensional modelling, advanced topics- data extraction, transformation and loading, data quality, OLAP in Data Warehouse, Data warehousing and the web. Implementation and Maintenance: Physical design process, Data Warehouse deployment, growth and maintenance.

UNIT 2: Introduction to Data Mining Introduction (8 Lectures)

Basics of data mining, Different definitions of Data Mining and related concepts, Data mining process, Data preparation, data cleaning and data visualization. KDD process, Data mining techniques: Clustering, Association rules and Decision trees.

UNIT 3: Clustering (15 Lectures)

Concept of Similarity and distance, Euclidean distance, Manhattan distance, Cosine similarity, Jaccard coefficient, Partitional versus Hierarchical Clustering, different types of data in clustering, Partitional clustering methods – k-means, k-medoids, PAM, CLARA, CLARANS. Hierarchical clustering methods – BIRCH, CURE, Density based clustering methods-DBSCAN.

UNIT 4: Rule Mining (15 Lectures)

What is an association rule? Mining association rules, frequent sets and border sets, algorithms for mining association rules – Apriori algorithm, Pincer-Search algorithm, Border algorithm.

UNIT 5: Classification (10 Lectures)

Introduction, Clustering versus Classification, decision tree construction principle, decision tree generation algorithms – CART, ID3.

REFERENCE BOOKS

1. A.K. Puzari, *Data Mining Techniques*, University Press.
2. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufman. 2001.
3. P. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, Pearson Education (LPE); 2009.

BCA-HE-5036: COMPUTER ORIENTED NUMERICAL METHODS AND STATISTICAL TECHNIQUES

(Credit: 4+2=6)(L: 4, P: 4, T: 0)
Theory: 60 Lectures, Practical: 20 Lectures

UNIT 1: Representation of numbers **(4 Lectures)**

Floating point representation, single and double precision, round off errors and truncation errors

UNIT 2: Solution of non-linear equation **(7 Lectures)**

Bisection method, Newtons method, Regula Falsi method.

UNIT 3: Solution of simultaneous linear equation **(12 Lectures)**

Basic elimination method, Gaussian elimination method, Gauss Jordan method, method of successive approximation.

UNIT 4: Ordinary differential equation **(6 Lectures)**

Euler's method, Runge Kutta method.

UNIT 5: Interpolation **(8 Lectures)**

Newton's interpolation, Lagrange's interpolation, Newton's divided difference method.

UNIT 6: Numerical integration **(11 Lectures)**

Trapezoidal rule, Simpson's 1/3rd and Simpson's 3/8th rule.

UNIT 7: Statistical methods **(12 Lectures)**

Measure of central tendency: Mean, Median and Mode, Probability, probability distribution, Binomial, Poison and normal distribution, Mathematical expectations, moments, correlation, regression.

REFERENCE BOOKS

1. M.K.Jain, S.R.K.Iyenger, R.K.Jain, — *Numerical methods for Scientific and Engineering Computation*, Wiley Easterns.
2. K.E. Atkinson, —*An introduction to numerical analysis*, J.Willey and Sons.

Practical / Lab work to be performed

(N.B: Student has to perform **any six** of the following experiments**)**

1. Find the roots of the equation by bisection method.
2. Find the roots of the equation by Regula–Falsi method.
3. Find the solution of a system of nonlinear equation using Newton's method.
4. Find the solution of simultaneous linear equations using Gauss Elimination method.

5. Find the solution of system of equations using Gauss-Jordan method.
6. Evaluate the approximate value of finite integrals using Simpson's 1/3rd and Simpson's 3/8th rule.
7. Implement Runge Kutta method for ordinary differential equations.
8. Implement Newton's interpolation method.

Note: Programming is to be done in any one of Computer Algebra Systems:
MATLAB / MATHEMATICA / MAPLE.

Reference Books

1. Laurence V. Fausett, Applied Numerical Analysis, Using MATLAB, Pearson, 2/e (2012)
2. M.K. Jain, S.R.K. Iyengar and R.K. Jain, Numerical Methods for Scientific and Engineering Computation, New Age International Publisher, 6/e (2012)
3. Steven C Chapra, Applied Numerical Methods with MATLAB for Engineers and Scientists, Tata McGraw Hill, 2/e (2010)

BCA-HE-5046: PROGRAMMING IN PYTHON

(Credit: 4+2=6)(L: 4, P: 4, T: 0)

Theory: 60 Lectures Practical: 60 Lectures

UNIT 1: Planning the Computer Program (4 Lectures)

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation

UNIT 2: Techniques of Problem Solving (6 Lectures)

Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming

UNIT 3: Overview of Programming (4 Lectures)

Structure of a Python Program, Elements of Python

UNIT 4: Introduction to Python (6 Lectures)

Python Interpreter, Using Python as calculator, Python shell, Indentation Atoms, Identifiers and keywords, Literals, Strings, Operators(Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator)

UNIT 5: Creating Python Programs (8 Lectures)

Input and Output Statements, Control statements (Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass), Defining Functions, default arguments, Errors and Exceptions

UNIT 6: Iteration and Recursion (8 Lectures)

Conditional execution, Alternative execution, Nested conditionals, the return statement, Recursion, Stack diagrams for recursive functions, Multiple assignment, the while statement, Tables, Two-dimensional tables

UNIT 7: Strings and Lists (8 Lectures)

String as a compound data type, Length, Traversal and the for loop, String slices, String comparison, A find function, Looping and counting, List values, Accessing elements, List length, List membership, Lists and for loops, List operations, List deletion. Cloning lists, Nested lists

UNIT 8: Object Oriented Programming (4 Lectures)

Introduction to Classes, Objects and Methods, Standard Libraries

UNIT 9: Data Structures (6 Lectures)

Arrays, list, set, stacks and queues.

UNIT 10: Searching and Sorting (6 Lectures)

Linear and Binary Search, Bubble sort, Selection sort and Insertion sort.

Practical / Lab work to be performed

(N.B: Student has to perform **any 12** of the following experiments**)**

1. Using for loop, print a table of Celsius/Fahrenheit equivalences. Let c be the Celsius temperatures ranging from 0 to 100, for each value of c, print the corresponding Fahrenheittemperature.
2. Using while loop, produce a table of sins, cosines and tangents. Make a variable x in range from0 to 10 in steps of 0.2. For each value of x, print the value of sin(x), cos(x) and tan(x).
3. Write a program that reads an integer value and prints ―leap year‖ or ―not a leap year.
4. Write a program that takes a positive integer n and then produces n lines of output shown asfollows. For example enter a size: 5

```
*
**
***
****
*****
```

5. Write a function that takes an integer n as input and calculates the value of 1 + 1/1! + 1/2! + 1/3! + … + 1/n
6. Write a function that takes an integer input and calculates the factorial of that number.
7. Write a function that takes a string input and checks if it is a palindrome or not.
8. Write a list function to convert a string into a list, as in list ('abc') gives [a, b, c].
9. Write a program to generate Fibonacci series.
10. Write a program to check a number is Armstrong or not
11. Write a program to check whether the input number is even or odd.
12. Write a program to print all even number between a range(for example between 1 and100).
13. Write a program to print all prime number between a range(for example between 1 and100).
14. Write a program to compare three numbers and print the largest one.
15. Write a program to print factors of a given number.
16. Write a method to calculate GCD of two numbers.
17. Write a program to create Stack Class and implement all its methods. (Use Lists).
18. Write a program to create Queue Class and implement all its methods. (Use Lists)
19. Write a program to implement linear and binary search on lists.
20. Write a program to sort a list using insertion sort and bubble sort and selection sort.

REFERENCE BOOKS

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. Python Tutorial/Documentation www.python.or 2015
3. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist :learning with Python , Freely available online.2012
4. http://docs.python.org/3/tutorial/index.html
5. http://interactivepython.org/courselib/static/pythonds
6. http://www.ibiblio.org/g2swap/byteofpython/read/