VIJAYANAGARA SRI KRISHNADEVARAYA UNIVERSITY

JNANASAGARA CAMPUS, BALLARI-583105

Department of Computer Science

SYLLABUS

Bachelor of Computer Applications

(II Semester)

With effect from

2021-22

VIJAYANAGARA SRI KRISHNADEVARAYA UNIVERSITY

Department of Computer Science

Jnana Sagara, Ballari - 583105

Distribution of Courses/Papers in Undergraduate Programme II Semester as per Choice Based Credit System (CBCS) Proposed for UG (BCA) Programs

II-SEMESTER

FIRST YEAR; SEMESTER-2										
Objective: Understanding, Exploration & Ability to solve well defined problems										
Category	**Course code**	**Title of the Paper**	**Marks**			**Teaching hours/week**			**Credit**	**Duration of exams (Hrs)**
			IA	**SEE**	**Total**	**L**	**T**	**P**		
L-3	21BCA2L3LK2	Kannada	40	60	100	4	0	0	3	3
	21BCA2L3FKL2	Functional Kannada								
L-4	21BCA2L4EN2	English	40	60	100	4	0	0	3	3
	21BCA2L4HI2	Hindi								
	21BCA2L4SN2	Sanskrit								
	21BSC2L4TE2	Telugu								
	21BCA2L4UR2	Urdu								
DSC4	21BCA2C4L	Data Structures using C	40	60	100	3	0	0	3	3
	21BCA2C4P	Data Structures Lab	25	25	50	0	0	4	2	3
DSC5	21BCA2C5L	Object Oriented Concepts using Java	40	60	100	3	0	0	3	3
	21BCA2C5P	Java Lab	25	25	50	0	0	4	2	3
DSC6	21BCA2C6L	Discrete Mathematics Structures	40	60	100	3	0	0	3	3
OEC2	Students should opt other Department Open Elective Course.		40	60	100	3	0	0	3	3
AECC1	21BCA2AE1L	Environmental Studies	20	30	50	2	0	0	2	1
VBC3	21BCA2V3PE2	Physical Education – Sports	-	-	25	-	-	2	1	-
VBC4	21BCA2V4NC1	NCC/NSS/R&R(S&G)/Cultural/RedCross	-	-	25	-	-	2	1	-
			Total Marks		**850**	**Semester Credits**			**26**	

Course Title: Data Structures using C	Course code: 21BCA2C4DSL
Total Contact Hours: 42	Course Credits: 03
Formative Assessment Marks: 40	Duration of ESA/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (CO's):

At the end of the course, students will be able to:

1. Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms
2. Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs
3. Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs
4. Demonstrate different methods for traversing trees
5. Compare alternative implementations of data structures with respect to performance
6. Describe the concept of recursion; give examples of its use.
7. Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing

DSC4: Data Structures using C

Unit	Description	Hours
1	**Introduction to data structures:** Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures. Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - malloc, calloc, realloc and free. Algorithm Specification, Performance Analysis, Performance Measurement Recursion: Definition; Types of recursions; Recursion Technique Examples - GCD, Binomial coefficient nCr, Towers of Hanoi; Comparison between iterative and recursive functions.	08
2	**Arrays**: Basic Concepts – Definition, Declaration, Initialization, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory. Traversing linear arrays; Inserting and deleting elements; Sorting – Selection sort, Bubble sort, Quick sort, Selection sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching; Multidimensional arrays; Representation of multidimensional arrays; Sparse matrices.	10
3	**Linked list:** Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly liked list, Header liked list, Circular linked list; Representation of Linked list in Memory;	08

	Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection,	
4	**Stacks:** Basic Concepts – Definition and Representation of stacks; Operations on stacks; Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack; Application of stack in function calls. **Queues**: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues	08
5	**Trees:** Definition; Tree terminologies –node, root node, parent node, ancestors of a node, siblings, terminal & non-terminal nodes, degree of a node, level, edge, path, depth. **Binary tree**: Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array representation of binary tree. Traversal of binary tree; preorder, In-order and Post-order traversal.	08

References:

1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures
2. Tanenbaum: Data structures using C (Pearson Education)
3. Kamathane: Introduction to Data structures (Pearson Education)
4. Y. Kanitkar: Data Structures Using C(BPB)
5. Kottur: Data Structure Using C
6. Padma Reddy: Data Structure Using C
7. Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions (McGraw Hill Education,2007)

Course Title: Object Oriented Concepts using JAVA	Course code: 21BCA2C5OJL
Total Contact Hours: 42	Course Credits: 03
Formative Assessment Marks: 40	Duration of ESA/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (CO's):

At the end of the course, students will be able to:

1. Understand the features of Java and the architecture of JVM
2. Write, compile, and execute Java programs that may include basic data types and control flow constructs and how typecasting is done
3. Identify classes, objects, members of a class and relationships among them needed for a specific problem and demonstrate the concepts of polymorphism and inheritance
4. The students will be able to demonstrate programs based on interfaces and threads and explain the benefits of JAVA's Exceptional handling mechanism compared to other Programming Language
5. Write, compile, execute Java programs that include GUIs and event driven programming and also programs based on files

DSC5: Object Oriented Concepts using Java

Unit	Description	Hours
1	**Introduction to Java:** Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java	06
2	**Objects and Classes:** Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, String Buffer, File, this reference	06
3	**Inheritance and Polymorphism:** Inheritance in java, Super and sub class, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, UTIL package.	08
4	**Event and GUI programming:** Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle, Introduction to swing, Exceptional handling mechanism.	10
5	**I/O programming:** Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files.	06
6	**Multithreading in java:** Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try catch-finally,	06

	Collections in java, Introduction to JavaBeans and Network Programming	

References:

1. Programming with Java, By E Balagurusamy – A Primer, Fourth Edition, Tata McGraw Hill Education Private Limited.
2. Core Java Volume I – Fundamentals, By Cay S. Horstmann, Prentice Hall
3. Object Oriented Programming with Java :Somashekara M.T., Guru, D.S., Manjunatha K.S
4. Java 2 - The Complete Reference – McGraw Hill publication.
5. Java - The Complete Reference, 7th Edition, By Herbert Schildt– McGraw Hill publication.

Course Title: Discrete Mathematical Structures	Course code: 21BCA2C6DML
Total Contact Hours: 42	Course Credits: 03
Formative Assessment Marks: 40	Duration of ESA/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (CO's):

At the end of the course, students will be able to:

1. To understand the basic concepts of Mathematical reasoning, set and functions.
2. To understand various counting techniques and principle of inclusion and exclusions.
3. Understand the concepts of various types of relations, partial ordering and Equivalence relations.
4. Apply the concepts of generating functions to solve the recurrence relations.
5. Familiarize the fundamental concepts of graph theory and shortest path algorithm

DSC6: Discrete Mathematical Structures

Unit	Description	Hours
1	**The Foundations**: Logic and proofs: Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy. **Basic Structures**: Sets, Functions, Sequences, Sums, and Matrices: Sets, set operations, Functions, Sequences and Summations, matrices.	12
2	**Counting**: Basics of counting, Pigeonhole principle, Permutation and combination, Binomial Coefficient and Combination, Generating Permutation and Combination. **Advanced Counting Techniques**: Applications of Recurrence Relations, Solving Linear Recurrence, Relations, Divide and Conquer Algorithms and Recurrence Relations, Generating functions, Inclusion-Exclusion, Applications of Inclusion-exclusion.	10
3	**Induction and Recursion**: Mathematical Induction, Strong Induction and Well Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Corrections. **Relation**: Properties of relation, Composition of relation, Closer operation on relation, Equivalence relation and partition. Operation on relation, Representing relation.	12
4	**Graphs**: Graphs and Graph models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring	08

References:

1. Discrete Mathematics and Its Applications, Kenneth H. Rosen: Seventh

Edition, 2012.
2. Discrete Mathematical Structure, Bernard Kolman, Robert C, Busby, Sharon Ross, 2003.
3. Graph Theory with Applications to Engg and Comp. Sci: Narsingh Deo-PHI 1986
4. Discrete and Combinatorial Mathematics Ralph P. Grimaldi, B. V. Ramatta, Pearson, Education, 5th Edition
5. Discrete Mathematical Structures, Trembley and Manobar

Course Title: **Data Structures Lab**	Course code: 21BCA2C4DSP
Total Contact Hours: 52	Course Credits: 02
Formative Assessment Marks: 25	Duration of ESA/Exam: 03 Hours
Summative Assessment Marks: 25	

Course Outcomes (CO's):

At the end of the course, students will be able to:

1. To implement recursive functions.
2. To arrange data using different sorting techniques.
3. To implement stack, queue, linked list and tree.

Data Structures Lab

Programming Lab

Part A:

1. Program to find GCD using recursive function
2. Program to display Pascal Triangle using binomial function
3. Program to generate n Fibonacci numbers using recursive function.
4. Program to implement Towers of Hanoi.
5. Program to implement dynamic array, find smallest and largest element of the array.
6. Program to create two files to store even and odd numbers.
7. Program to create a file to store student records.
8. Program to read the names of cities and arrange them alphabetically.
9. Program to sort the given list using selection sort technique.
10. Program to sort the given list using bubble sort technique.

Part B:

1. Program to sort the given list using insertion sort technique.
2. Program to sort the given list using quick sort technique.
3. Program to sort the given list using merge sort technique.
4. Program to search an element using linear search technique.
5. Program to search an element using recursive binary search technique.
6. Program to implement Stack.
7. Program to convert an infix expression to postfix.
8. Program to implement simple queue.
9. Program to implement linear linked list.
10. Program to display traversal of a tree.

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part A	Writing the Program	05
	Execution	05
Program -2 from Part B	Writing the Program	05
	Execution	05
Viva Voice		05
Total		25

Course Title: JAVA Lab	Course code: 21BCA2C5OJP
Total Contact Hours: 52	Course Credits: 02
Formative Assessment Marks: 25	Duration of ESA/Exam: 03 Hours
Summative Assessment Marks: 25	

Course Outcomes (CO's):

At the end of the course, students will be able to:

1. Implement Object Oriented programming concept using basic syntaxes of control Structures
2. Identify classes, objects, members of a class and the relationships among them needed for a finding the solution to specific problem
3. Demonstrates how to achieve reusability using inheritance
4. Demonstrate understanding and use of interfaces, packages, different exception handling mechanisms and concept of multithreading for robust faster and efficient application development.
5. Identify and describe common user interface components to design GUI in Java using Applet & AWT along with response to events

JAVA Lab

Practice Lab

1. Program to print the following triangle of numbers 1

   1 2

   1 2 3

   1 2 3 4

   1 2 3 4 5
2. Program to simple java application, to print the message, "Welcome to java"
3. Program to display the month of a year. Months of the year should be held in an array.
4. Program to find the area of rectangle.
5. program to demonstrate a division by zero exception
6. Program to create a user defined exception say Pay out of Bounds.

Programming Lab

PART A: Java Fundamentals OOPs in Java

1. Program to assign two integer values to X and Y. Using the 'if' statement the output of the program should display a message whether X is greater than Y.
2. Program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop. (Hint Fact of 4 =4*3*2*1)
3. Program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.
4. Program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from Add, Sub class to use the member data of the super class. MulDiv should have methods to multiply and divide A main function should access the methods and perform the mathematical operations.
5. Program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.
6. Program
   a. To find the area and circumference of the circle by accepting the radius from the user.
   b. To accept a number and find whether the number is Prime or not
7. Program to create a student class with following attributes.
   Enrollment No: Name, Mark of sub1, Mark of sub2, mark of sub3, Total Marks. Total of the three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50. If a candidate fails in any one of the subjects his total mark must be declared as zero. Using this condition write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.
8. In a college first year class are having the following attributes Name of the class (BCA, BCom, BSc), Name of the staff No of the students in the class, Array of students in the class
9. Define a class called first year with above attributes and define a suitable constructor. Also write a method called best Student () which process a first-year object and return the student with the highest total mark. In the main method define a first-year object and find the best student of this class
10. Program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment. ie, print them as

per their seniority.

11. Create a package‘ student. Fulltime BCA ‘in your current working directory
    a. Create a default class student in the above package with the following attributes: Name, age, sex.
    b. Have methods for storing as well as displaying

PART B: Exception Handling & GUI Programming

1. Program to catch Negative Array Size Exception. This exception is caused when the array is initialized to negative values.
2. Program to handle NullPointerException and use the “finally” method to display a message to the user.
3. Program which create and displays a message on the window
4. Program to draw several shapes in the created window
5. Program to create an applet and draw gridlines
6. Program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother similar details of mother also appear.
7. Create a frame which displays your personal details with respect to a button click
8. Create a simple applet which reveals the personal information of yours.
9. Program to move different shapes according to the arrow key pressed.
10. Program to create a window when we press M or m the window displays Good Morning, A or the window displays Good After Noon E or e the window displays Good Evening, N or n the window displays Good Night
11. Demonstrate the various mouse handling events using suitable example.
12. Program to create menu bar and pull-down menus.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part A	Writing the Program	05
	Execution	05
Program -2 from Part B	Writing the Program	05
	Execution	05
Viva Voice		05
Total		25