

LAB MANUAL

Bachelor of Computer Application

BCA

(2024-2027) onwards

Vision

To develop the Department of Computer Science & Information Technology as a Centre for Excellence to produce leading Professionals who can serve the society with innovative skills, Computer Experts, Researchers to meet the needs of the software industry in national /global scenario responding to the challenges of ever changing world.

Mission

- We endeavour to provide the best possible learning environment to enhance innovations, research capabilities, problem solving skills, leadership qualities, team spirit and ethical responsibilities.
- To nurture the talent of the students to be successful, ethical and effective problem solvers who will contribute positively to the economic growth of the nation and prepare to respond to the challenges.

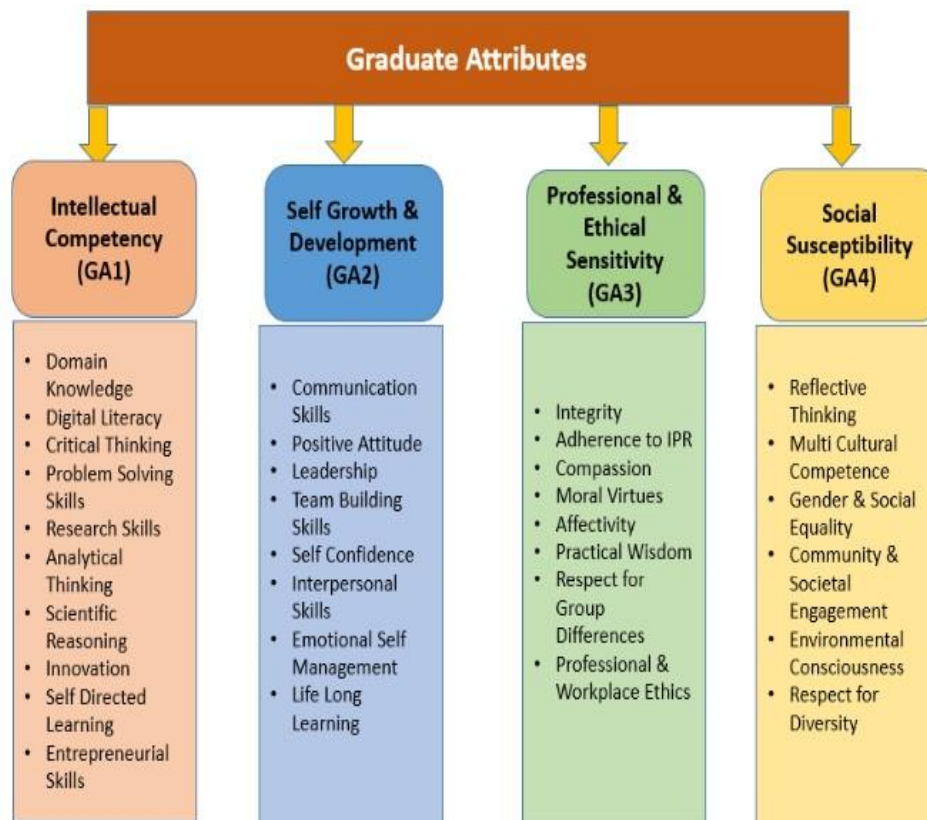


Graduate Attributes

Jharkhand Rai University is a mecca of transformative education which strongly believes in the holistic development of students. The university provides the cutting-edge of holistic learning to develop promising youngsters into leaders of tomorrow with globally relevant, future-ready and actionable intelligence. The objective of the Department is to make each student proficient in synthesizing/analysing information and be ethical, socially responsible, and just when making decisions. JRU ensures inclusive and equitable quality education and promote lifelong learning opportunities for all.

Every graduate of the Department will be developed to possess the following attributes:

1. Intellectual Competency
2. Self-Growth & Development
3. Professional & Ethical Sensitivity
4. Social Susceptibility





Program: BCA

Semester: First

Course: Computer Fundamentals Lab

Course Code: 3CCC101P

L	T	P	C
0	0	2	1

Course Objective:

Students will be able to

1. To understand basic of computer and working with OS.
2. To develop working skill with productivity tools, graphics designing and internet.
3. To acquire basic programming skills.
4. To apply computing in problem solving.

Course Outcome:

After the successful completion of the course, the students will be able to:

1. Converse in Basic Computer Terminology.
2. Formulate opinions about the impact of computers on society.
3. Process the knowledge of basic hardware peripherals.
4. Bridge the fundamental concepts of computers with the present level of knowledge of the students.

Part A: Hardware

1. Identification of the peripherals of a computer, components in a CPU and their functions.
2. Assembling and disassembling the system hardware components of personal computer.
3. Basic Computer Hardware Trouble shooting.
4. LAN and WiFi Basics.
5. Operating System Installation – Windows OS, UNIX/LINUX, Booting.
6. Installation and Uninstallation of Software – Office Tools, Utility Software (like Anti-Virus, System Maintenance tools); Application Software - Like Photo/Image Editors, Audio Recorders/Editors, Video Editors...); Freeware, Shareware, Payware and Trialware; Internet Browsers, Programming IDEs,

Part B: Software

1. Activities using Word Processor Software
2. Activities using Spreadsheets Software
3. Activities using Presentation Software
4. Activities involving Multimedia Editing (Images, Video, Audio ...)
5. Tasks involving Internet Browsing

NOTE: In addition to the ones listed above, universities can include other activities so as for the student to become proficient in using personal computers for multiple purposes for which modern computers can be put to use.

Reference:

1. Computational Thinking for the Modern Problem Solver, By Riley DD, Hunt K.A
CRCpress, 2014
2. Ferragina P, Luccio F. Computational Thinking: First Algorithms, Then Code. Springer

Web References:

<http://www.flowgorithm.org/documentation/>



Program: BCA

Semester: First

Course: C Programming Lab

Course Code: 3CCC102P

L	T	P	C
0	0	2	1

Course Objective:

Students will be able to

1. Construct programs that demonstrate effective use of advanced c features including the pre-processor, pointers, void, static and external variables, advanced data structures, and dynamic memory management
2. Select and model data using primitive and structured types
3. Analyze and construct effective algorithms
4. Use development environment features including make processors, editors, debuggers, compilers, linkers, and libraries.
5. Identify and comprehend c documentation
6. Work well with peer developers in a team situation including mentoring and peer reviews

Course Outcome:

After the successful completion of the course, the students will be able to:

1. Use the fundamentals of C programming in trivial problem solving
2. Identify solution to a problem and apply control structures and user
3. Defined functions for solving the problem
4. Apply skill of identifying appropriate programming constructs for problem
5. solving



Course Content:

Program:

1. To print the word “Hello”.
2. To sum two nos. Where the numbers are taken through the keyboard.
3. To calculate Simple Interest where the principle, no. Of years and rate are given.
4. An employee basic salary is less than Rs. 1500, then HRA = 10% of basic salary and DA = 90% OF BASICS. If this salary is either equal to or above Rs. 1500 then HRA = Rs. 500 and DA = 98% OF BASIC SALARY. If the employee’s salary is input through the keyboard then write a program to find its Gross Salary, where Gross Salary = Basic Salary + HRA + DA
5. The marks obtained by a student in 5 different subjects are input through the keyboard. The student gets a division as per the following rules:

Percentage above or equal to 60 – First Division
Percentage between 50 and 59 – Second Division
Percentage between 40 and 49 – Third Division
Percentage less than 40 – Fail

6. To print numbers from 1 to 10 using for loop
7. Print: 1 2 3 4 4 3 2 1
1 2 3 and 4 3 2
1 2 4 3
1 4
8. To find the factorial of a given number.
9. To print Fibonacci numbers.
10. To display reversing digit of an integer.
11. To demonstrate switch case.
12. To calculate the sum of three number using function.
13. To find the square of any number using function.
14. To swap two numbers using function.
15. To find the sum of 20 numbers using an array.
16. To print a 2D – array.



Program: BCA

Semester: Second

Course: Data Structure Using C Lab

Course Code: 3CCC103P

L	T	P	C
0	0	2	1

Course Objective:

Students will be able to

- To develop skills to design and analyze simple linear and nonlinear data structures
- To Strengthen the ability to identify and apply the suitable data structure for the given real world problem
- To design and analyze the time and space efficiency of the data structure
- To Gain knowledge in practical applications of data structures

Course Outcome:

After the successful completion of the course, the students will be able to:

- Understand basic data structures such as arrays, linked lists, stacks and queues.
- Describe the hash function and concepts of collision and its resolution methods
- Solve problem involving graphs, trees and heaps
- Apply Algorithm for solving problems like sorting, searching, insertion and deletion of data

Course Content:

Program:

1. To read and display n numbers using an array.
2. To find transpose a 3 X 3 matrix.
3. To insert a number at a given location in an array.
4. To delete a number from a given location in an array.
5. To create a linked list and perform insertions
 - a) At beginning
 - b) At end
 - c) Before a given node
 - d) After a given node
6. To create a linked list and perform deletions
 - a) From beginning
 - b) From end
 - c) At a given node
 - d) After a given node
7. To create a circular linked list and perform insertion at the beginning of list.
8. To create a circular linked list and perform insertion at the end of list.
9. To create a circular linked list and perform deletion from the beginning of list.
10. To create a circular linked list and perform deletion from the end of list.
11. To perform Push, Pop and Peep operations on a stack.
12. To implement a linear queue.
13. To implement a linked queue.
14. To implement a priority queue.
15. To implement a Binary Search tree and perform the following:
 - a) Insert Element b) Preorder Traversal
16. To implement a Binary Search tree and perform the following:
 - a) Insert Element b) Post-order Traversal
17. To search an element in an array using linear search technique.
18. To search an element in an array using binary search technique.
19. To sort an array using insertion sort algorithm.
20. To implement quick sort algorithm.

Program: BCA

Semester: Second

Course: JAVA Programming Lab

Course Code: 3CCC104P

L	T	P	C
0	0	2	1

Course Objective:

Students will be able to

1. Understand the basics of JAVA programs and its execution.
2. Differentiate between C++ and Java programming.
3. Concepts like packages and interfaces.
4. Understand life cycle of the applets and its functionality.
5. Understand the usage util package.
6. Develop java programs using interfaces

Course Content:

Program:

1. To find square root of given number
2. To enter principal, rate & time and find simple interest
3. To find whether a year is leap year or not
4. To enter a number from keyboard and find out Fibonacci series
5. To enter a number from keyboard and find out factorial of the number
6. To enter a number from keyboard and check whether the number is palindrome or not
7. To enter a number from keyboard and print the prime numbers present within it
8. To enter a number from keyboard and determine whether it is Armstrong or not.
9. Program to demonstrate switch statement
10. To swap two numbers without using third variable
11. To find the greatest among 3 numbers
12. Program to sort an array in an ascending order
13. Program to find out the sum and average of the elements present in an array
14. Program to add the elements of two different two dimensional array.
15. Program to find out the biggest and smallest number from a matrix.
16. To implement the concept of final class
17. To implement the concept of interface
18. Program to reverse a specified string.
19. Showing a program using package.
20. To create an applet
21. To implement the concept of thread

Program: BCA
Semester: Second
Course: Computer Assembly and Repair
Course Code: 3CSEC102

L	T	P	C
1	0	2	2

1. Demonstration of Hardware peripherals: CPU, RAM, SMPS, Motherboard, NIC card, Processor, Processor cooling fan, PCI card, HDD.
2. Demonstration of various ports: CPU , VGA port, PS/2 (keyboard, mouse) ,USB, LAN, Speaker, Audio.
3. Identify the Computer Name and Hardware Specification (RAM capacity, Processor type, HDD, 32 bit/ 64 bit)
4. Identify and Troubleshoot the problems of RAM (beep sound with blue screen), SMPS and motherboard (CPU is not switched ON)
5. Configure BIOS settings- disable and enable USB and LAN.
6. Identify, how to recover the hidden files from corrupted pendrive using command.
7. Recover the contents from crashed Hard Disk using Disk Drill software.
8. Install Operating System – Windows family (Windows 7/ Windows 10) and also make partitions.
9. Install Operating System - Unix family (Linux/UBUNTU)
10. Install Application software – python 3.8, MS- Office 2010/2013, MySQL, TOAD, Openoffice, etc.,
11. Install any one of the antivirus software (Avast, Kaspersky, etc.,) and observe the variations before and after installation.
12. Add new Hardware device (keyboard, mouse, Speaker, Microphone)
13. Connect the LCD Projector with Laptop / CPU.
14. Adding additional RAM to the system.(expanding RAM size).
15. Graphic Card insertion.
16. Assemble and Disassemble Desktop System.

References:

1. Dan Gookin ,Troubleshooting & Maintaining Your PC ALL-IN-ONE, 3rd Edition,2017, John Wiley & Sons.
2. Mike Meyers, Scott Jernigan, Dan Lachance, ”CompTIA Fundamentals + Exam Guide (All-in-One), 2nd Edition, 2019, Mc Graw Hill Education.

Web References:

1. https://www.youtube.com/watch?v=ItxwyMR0SnY&list=PLeH4ngtDM7eE-1_mdWuXWyZrI_FMHnyJ0&index=5
2. <https://www.cleverfiles.com/howto/crashed-hard-drive-recovery.html>

Program: BCA

Semester: Third

Course: Python Programming Lab

Course code: 3CCC201P

L	T	P	C
0	0	2	1

Course Objective:

Students will be able to

1. To learn and understand Python programming basics and paradigm.
2. To learn and understand python looping, control statements and string manipulations.
3. Students should be made familiar with the concepts of GUI controls and designing GUI applications.
4. To learn and know the concepts of file handling, exception handling and database connectivity.

List of experiments:

1. Write a program to calculate the average of three subject's mark of a student.
2. Write a program to swap two numbers.
3. Write a python program to convert temperature from Celsius to Fahrenheit.
4. Write a python program to find largest among three numbers
5. Write a python program to print all the prime numbers less than 20.
6. Generate Fibonacci series up to n term.
7. Write a program to display greatest common divisor of two numbers.
8. Enter any string and check whether it is palindrome.
9. Enter any string and count total number of vowel and consonant.
10. Write a python program to create, append and remove lists in python.
11. Write a program to demonstrate working with tuples in python.
12. Write a program to demonstrate working with dictionaries in python.
13. Write a python program to find factorial of a number using recursion.
14. Write a python program to define a module and import a specific function in that module to another program.
15. Display the average of three numbers using lambda function
16. Display the factorial of any number using tkinter.
17. Write a program for the crud operation.
18. Write a program using numpy module.
19. Write a program using panda module.
20. Write a program using matplotlib module.

Program: BCA

Semester: Third

Course: Database Management System Lab

Course code: 3CCC202P

L	T	P	C
0	0	2	1

Course Objective:

Students will be able to

- Understand, analyze and apply common SQL statements including DDL, DML and DCL statements to perform different operations.
- Design different views of tables for different users and to apply embedded and nested queries.
- Design and implement a database for a given problem according to well-known design principles that balance data retrieval performance with data consistency.



1. Database Schema for Emp_Details

Emp(Eid, Ename, Age, Dept, Salary, Adress, City, State)

For the above schema, perform the following —

- a) Create the table with the appropriate data types and integrity constraints
- b) Insert around 10 records in the **Emp** table
- c) List all the employee names whose age is 40 years.
- d) List all the employee names with their City name, whose salary is greater than 30000.
- e) List the details of employees whose basic salary is between Rs. 10,000 and 25,000.
- f) Display the names of only those employees who belong to Mumbai.
- g) List the details for an employee whose Eid=5
- h) Create a view which lists out the Eid, Ename and Dept.
- i) List the names of employees having salary greater than Rs. 10000 or first alphabet of his name is 'B'
- j) Delete the records of those employees whose salary is greater than Rs. 25000.

2. Database Schema for Std_Details

Std(Roll_no, Name, Age, Dept, Course, Semester, Fee, Result)

For the above schema, perform the following —

- a) Create the tables with the appropriate data types and integrity constraints
- b) Insert around 10 records in the **Std** table
- c) Create a view which lists out the Roll_no and Name.
- d) Create a view which will show the student names and their result.
- e) List the students who have exactly five letters in their name.
- f) Find out the total fee paid by the students.
- g) Create a view which will display the table in ascending order with the names of student.
- h) Display the names of students having at least five characters in their name.
- i) Add a penalty of Rs. 1500 in the fee of second student
- j) Create a view which will display the student details belonging to either Delhi, Mumbai or Kolkata.

3. Database Schema for Sports_Details Player(Reg_no, Name, Age, Sports_name, Fee)

For the above schema, perform the following-

- a) Create the tables with the appropriate data types and integrity constraints
- b) Insert around 10 records in the **Player** table.
- c) Change the name of the fifth player to 'Ramesh'
- d) List the players who play tennis or cricket.
- e) Create a view which will display the player names and fee in descending order with their names.
- f) Delete all the player's records whose age is below 20 years.
- g) Count total number of rows in the table.
- h) Add a new column **Address** to the existing table.
- i) Delete all the details from the table **Player**.
- j) Drop the table from its database.

4. Database Schema for Student_Library scenario

1. Std(Std_no, Std_name)

Membership(Mem_no, Std_no) Book(book_no, book_name, author) Iss_rec(iss_no, iss_date, Mem_no, book_no)

For the above schema, perform the following —

- a. Create the tables with the appropriate data type and integrity constraints
- b. Insert around 10 records in each of the tables
- c. List all the student names with their membership numbers
- d. Give a list of books taken by student with std_no as 7.
- e. List the detail of the students who borrowed book whose author is 'Floyd'
- f. List all the issues for the current date with student and book names
- g. List the book details which are issued as of today
- h. Give a count of how many books have been bought by each student
- i. Create a view which lists out the iss_no, iss_date, std_name, book name
- j. Create a view which lists the daily issues-date wise for the last one week.

5. Database Schema for a Customer_Sale scenario Customer(cust_id, cust_name)

Item(item_id, item_name, price) Sale(bill_no, cust_id, item_id, qty_sold)

For the above schema, perform the following —

- a) Create the tables with the appropriate data types and integrity constraints
- b) Insert around 10 records in each of the tables
- c) List the total Bill details with the quantity sold, price of the item and the final amount
- d) List the details of the customer who have bought a product which has a price>500
- e) List all the bills for the current date with the customer names and item numbers
- f) Give a count of how many products have been bought by each customer
- g) Give a list of products bought by a customer having cust_id as 5
- h) List the item details which are sold as of today
- i) Create a view which lists out the bill_no, bill_date, cust_id, item_id, price, qty_sold, amount
- j) Create a view which lists the daily sales date wise for the last one week



Program: BCA

Semester: Fourth

Course: Artificial Intelligence Lab

Course code: 3CCC204P

L	T	P	C
0	0	2	1

Course Objective:

Students will be able to

- Gain proficiency in various search algorithms, including heuristic and adversarial search techniques
- Understand the fundamental concepts and significance of Artificial Intelligence in various fields
- Understand the importance and methods of knowledge representation.
- To learn how to use the software python, prolog and LISP in AI.

Course Outcome:

After the successful completion of the course, the students will be able to:

- Solve AI problems using state space search techniques and various algorithms such as generate-and-test, hill climbing, and best-first search
- Develop simple knowledge-based systems to model real-world problems
- Implement basic NLP tasks such as syntactic parsing and semantic processing.
- Create basic expert systems for specific domains using expert system shells.
- Utilize tools and environments like Python programming, SWI-Prolog for logic programming, and CLISP for LISP programming.

List of experiments:

Unit I:

1.Production System: Water Jug Problem

Solve the Water Jug problem using a production system approach.

Software: Python

2.Heuristic Search: 8-Puzzle Problem

Implement the 8-puzzle problem using a heuristic search algorithm.

Software: Python

3.Hill Climbing: Find Local Maxima

Write a hill climbing algorithm to find local maxima in a mathematical function.

Software: Python

Unit II

4,Write a prolog program to represent few basic facts and perform queries (eg elephant is an animal. Elephant is bigger than horse etc.

5.Write a prolog program to represent few basic facts and perform queries (eg elephant is an animal. Elephant is bigger than horse etc.)

6.Predicate Logic: Representing Simple Facts

Create a program to represent simple facts using predicate logic.

Software: Prolog, SWI-Prolog

7.Predicate Logic: Family Tree Representation

Implement a family tree and represent relationships using predicate logic.

Software: Prolog, SWI-Prolog

Unit III:

8.Syntactic Processing: Sentence Parser

Write a program to parse simple sentences using a context-free grammar.

Software: Python, NLTK (Natural Language Toolkit)

Unit IV:

9.Expert System Shell: Simple Medical Diagnosis

Develop a simple expert system for medical diagnosis using an expert system shell.

Software: CLIPS, JESS

10. Expert System: Rule-Based Chatbot

Create a basic rule-based chatbot that responds to user queries.

Software: Python, AIML (Artificial Intelligence Markup Language)

11 Implement recursive functions in (e.g., factorial, Fibonacci).

Software: SWI-Prolog

Prolog: Solving Logic Puzzles

12. Use Prolog to solve simple logic puzzles (e.g., Sudoku).

Software: SWI-Prolog



Program: BCA

Semester: Fourth

Course: Computer Graphics and Multimedia Lab

Course code: 3CCC205P

L	T	P	C
0	0	2	1

Course Objective:

Students will be able to

- Demonstrate an understanding of contemporary graphics hardware.
- Create and formulate interactive graphics applications in programming language e.g C. Understand and explain the mathematical and theoretical principles of computer graphics. Eg: To draw basic objects like lines, triangles and polygons using built-in functions.
- Implement programs that demonstrate geometrical transformations.
- Understand about the modern software that are used in current scenario used in computer graphics.
- Model a simple multimedia application.



Program:

1. Program to draw a point
2. Program to draw line.
3. Program to draw a circle.
4. Program to draw a rectangle
5. Program to draw an arc.
6. Program to draw an ellipse. .
7. Program to draw a polygon.
8. Program to draw a filled polygon.
9. Program to draw a concentric circle.
10. Program to draw a sector.
11. Program to draw pie slice.
12. Program to draw a bar.
13. Program to draw a 3D bar.
14. Program to draw pie chart.
15. Program to draw bar chart.
16. Program to draw 3D Bar chart.
17. Program to draw a kite.
18. Program to draw a mesh ball.
19. Program to draw smiley face.
20. Program to draw a car.
21. To perform animation using any Animation software (Create Frame by Frame Animations using multimedia authoring tools)
22. To perform basic operations on image using any image editing software
23. To create a Jpeg image that demonstrates the various features of an image editing tool.