

DEPARTMENT OF COMPUTER APPLICATIONS

The Bachelor of Computer Applications (BCA) is a 3-year (6 semesters) undergraduate program affiliated to Bengaluru North University and Recognized by the Government of Karnataka. It caters to students with specific interests in computer science and information technology with computer science courses as the core and a wide range of electives. The course creates highly skilled, adaptable graduates who are able to design computer-based solutions to address information management and processing complications in industry, commerce, science, entertainment and the public sector.

The main focus of the Department of BCA is skilling the under graduates in disruptive technologies. The Department regularly conducts seminars, workshops and symposiums inviting the eminent speakers in industry to give a glimpse of latest advancements in industry.

PROGRAM EDUCATIONAL OBJECTIVES

PEO1: To equip the students to be updated with the Corporate world and Industry standard.

PEO2: To engage in professional development and to pursue post graduate education in the fields of Information Technology and Computer Applications.

PEO3: To mould the students not only in academics but also involve them in various technologies and updated courses within the college.

PROGRAM OUTCOMES (PO):

PO1: Understand the concepts of key areas in computer science.

PO2: Analyze and apply latest technologies to solve problems in the areas of computer applications.

PO3: To work effectively both as an individual and a team leader on multi disciplinary projects.

PO4: Apply technical and professional skills to excel in Industry or Corporate World.

PO5: Improves communication skills so that they can effectively present technical information in oral and written reports.

PO6: Develop practical skills to provide solutions to industry, society and business.

PO7: Prepares to create design innovative methodologies for solving complex-real life problems for the betterment of the society.

PO8: Gives overview of the topics in IT like networking, computer graphics, web development, trouble shooting, and hardware and software skills.

Course Outcomes (COs):

CO1: The curriculum prepares students for a career in software industry by equipping the students with the latest technology in order to be updated with IT field.

CO2: The program aims to educate students in Computer science and Information technology with emphasis on hands-on practical training in software development.

CO3: Creates the ability to design a computer application by considering realistic constraints some of which include safety, security and applicability.

CO4: Inculcating in students to work professionally pertaining to ethics, social, cultural and cyber regulations.

CO5: Involving students to be aware and handle any type of problem solving or practical implementation apart from computer related subjects which includes Mathematics, Accounting, and Soft skills.

Course Outcomes of BCA Offered by the Department of Computer Application

S.No	Name of the Paper	Program Specific Outcomes
1.	Unix Shell Programming	<p>Develop a strong ability in building programming concepts that supports file system, understand important aspects of shell programming, learn its architecture in depth as well the advantages of Unix operating system, process management.</p> <p>Understanding the Programming language to construct shell programming, filters, pattern matching, administration, process and file concepts in Unix operating system.</p>
2.	Data Communication and Networks	<p>Acquire knowledge and Practices of the different layers of OSI and TCP/IP models, paradigms and protocols, transmission mediums, peers.</p> <p>Learning concepts of communication network and security levels related to communication, LAN and MAC and various protocols functionality, various routers and switching techniques, and concepts of the various topologies along with their functionality.</p>
3.	Java Programming	<p>To differentiate between the concepts of OOPs and Java, Elements, browsers, JVM, Environment of Java. Understand the use of Applets and implementation of GUI in Programming Concepts.</p> <p>Key factors involved in multithreading, Interfaces, packages their creation and well as its utilization. Exceptions categories and its proper utilization in programming.</p>

S.No	Name of the Paper	Program Specific Outcomes
4.	Computer Architecture	<ul style="list-style-type: none"> Given the knowledge of digital circuits, and data structures, a Vth semester BCA student will be able to understand the basic principles on which computers work, analyze their performance and appreciate the issues affecting modern processors.
		<ul style="list-style-type: none"> Classify and compute the performance of machines. Understand how to implement memory chips, boards, modules and caches. Relate to arithmetic for ALU implementation. Understand the basics of hardwired and micro-programmed control of the CPU.
		<p>Identify, compare and assess issues related to ISA, memory, control and I/O functions. (Applying, Analyzing, Evaluating)</p> <ul style="list-style-type: none"> Estimate the performance of various classes of machines, memories, pipelined architectures etc. Compare CPU implementations, I/O methods etc. Analyze fast methods of ALU and FP unit implementations.
5.	Operating system	Describe and explain the fundamental components of a computer operating system.
		Define, restate, discuss, and explain the policies for scheduling, deadlocks, memory management, synchronization, system calls, and file systems.
		<p>Design and construct the following OS components: System calls, Schedulers, Memory management systems, Virtual Memory and Paging systems.</p> <p>Assessment: Students will design and implement the above OS components.</p>

6.	Database Management System	<p>Understand terms related to database design and management ☐</p> <p>Understand the objectives of data and information management . ☐</p> <p>Understand the database development process ☐</p> <p>Understand the relational model and relational database management system.</p>
		<p>Assess data and information requirements ☐</p> <p>Construct conceptual data models Develop logical data models.</p> <p>☐ Evaluate the normality of a logical data model, and correct any anomalies.</p> <hr/> <p>Implement relational databases using a RDBMS ☐</p> <p>Retrieve data using SQL ☐ Understand database performance issues ☐</p> <p>Understand the basics of data management and administration.</p> <hr/> <p>Understand the basics of data warehousing</p> <p>☐ Work as a valuable member of a database design and implementation team.</p>

7.	Data Structures	<p>Develop a strong knowledge of programming concepts, arrays, records, linked structure, stacks, queues, trees and graphs their representation in memory and uses by algorithm.</p> <hr/> <p>Develop hashing function for an application.</p> <p>Enhance programming in above concepts also dealing with sorting, searching and hashing and different traversal techniques.</p> <hr/> <p>Contrasting benefits of static and dynamic data structures</p>
----	------------------------	--

8.	Object Oriented Language UsingC++	Develop a strong ability in building programming concepts that supports efficient low-level computation, data abstraction, object-oriented programming, and generic programming language.
		Understanding the Programming language to construct and use new types of objects, virtual functions that match the concepts of computerapplication.
9.	Visual Programming	Acquire knowledge and Practices in visual programming learning (VPL) environment through experiments.
		Understand the difference between a console program and a Graphical User Interface (GUI) . To learn the application of a modern Integrated Development Environment (IDE) by describing the major components of an IDE and their functions.
10.	Software Engineering	Acquire the principle knowledge about the competence in designing, evaluating, and adapting software processes and software development tools to meet the needs of an advanced development project.
		Advance learning techniques andskills in eliciting user needs and designing an effective software solution.
11.	Microprocessor & Assembly level Language	To gain the general architecture of a microcomputer system and architecture of 8085 Microprocessor and understand the difference between 8085 and advanced microprocessor.
12.	Web Programming	To learn HTML tags and JavaScript Language programming concepts and techniques. To develop the ability to logically plan and develop web pages. To learn to write, test, and debug web pages using HTML and JavaScript.

13.	Discrete Mathematics	Develop logical thinking and its application to computer science to emphasize the importance of proving statements.
		Understand the concepts of mathematical logic for analyzing propositions and proving theorems.
14.	Digital Electronics	Understand the different types of digital electronic circuit using various mapping and mathematical methods.
		Identify the various digital ICs and understand their operation.
15.	Numerical And Statistical Methods	Understand the numerical and statistical methods to analyze, design and problem solving in the field of information technology.
		Develop numerical skills in solving the problems involving systems of linear equations, fitting of polynomials and different types of equations to the experimental data, derivatives and integrals.
16.	Operation Research	Understand the mathematical tools that are needed to solve optimisation problems.
		Able to design and solve simple models of CPM (Critical Path Method) to improve decision making and develop critical thinking and objective analysis of decision problems.
17.	Cryptography And Network Security	Understand the most common type of cryptographic algorithm, Public-Key Infrastructure and security protocols for protecting data on networks.
		Describe network security services and mechanisms, various network security, IPSec, Web Email security Malicious software.
18.	System programming	Describe the various phases of compiler and compare its working with assembler. Demonstrate how linker and loader create an executable program from an object module created by assembler and compiler.
		Understand the concepts and theory behind the implementation of high level programming Languages.

<p>19.</p>	<p>Theory Of Computation</p>	<p>The main benefit of theory of change comes from making different views and assumptions about the change process explicit, especially seemingly obvious ones. A good theory of change can specify how to create a range of conditions that help programs deliver on the desired outcomes.</p>
		<p>Discuss key notions of computation, such as algorithm, computability, decidability, reducibility, and complexity, through problem solving. Explain the models of computation, including formal languages, grammars and automata, and their connections.</p>
		<p>Analyze and design finite automata, pushdown automata, Turing machines, formal languages, and grammars. Solve computational problems regarding their computability and complexity and prove the basic results of the theory of computation.</p>
<p>20.</p>	<p>Problem Solving Techniques Using C</p>	<ul style="list-style-type: none"> • Able to implement the algorithms and draw flowcharts for solving Mathematical and Engineering problems • Demonstrate an understanding of computer programming language concepts.
		<ul style="list-style-type: none"> • To be able to develop C programs on linux platform. • Ability to design and develop Computer programs, analyzes, and interprets the concept of pointers, declarations, initialization, operations on pointers and their usage. • Able to define data types and use them in simple data processing applications also he/she must be able to use the concept of array of structures.

		<ul style="list-style-type: none">• Student must be able to define union and enumeration user defined data types.• Develop confidence for self education and ability for life-long learning needed for Computer language.
--	--	--