GOVERNMENT DEGREE COLLEGE
NAGARI
Department of Computer Science
COURSE OUTCOMES (2020)

1.	**3-1-108R**	**Problem Solving in C**	Upon successful completion of the course, a student will be able to: **CO1:** Understand the evolution and functionality of a Digital Computer. **CO2:** Apply logical skills to analyze a given problem **CO3:** Develop an algorithm for solving a given problem. **CO4:** Understand „C" language constructs like Iterative statements, Array processing, Pointers, etc. **CO5:** Apply "C" language constructs to the algorithms to write a "C" language program.
2.		**DATA STRUCTURES USING C**	Upon successful completion of the course, a student will be able to: **CO1:** Understand available Data Structures for data storage and processing. **CO2:** Comprehend Data Structure and their real-time applications - Stack, Queue, Linked List, Trees and Graph **CO3:** Choose a suitable Data Structures for an application **CO4:** Develop ability to implement different Sorting and Search methods **CO5:** Have knowledge on Data Structures basic operations like insert, delete, search, update and traversal **CO6:** Design and develop programs using various data structures **CO7:** Implement the applications of algorithms for sorting, pattern matching etc
3.		**Database Management Systems**	On completing the subject, students will be able to: **CO1:** Gain knowledge of Database and DBMS. **CO2:** Understand the fundamental concepts of DBMS with special emphasis on relational data model.

			CO3: Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database **CO4:** Model database using ER Diagrams and design database schemas based on the model. **CO5:** Create a small database using SQL. **CO6:** Store, Retrieve data in database.
4.		**Object orientated programming through Java**	At the end of this course student will: **CO1:** Understand the benefits of a well-structured program **CO2:** Understand different computer programming paradigms **CO3:** Understand underlying principles of Object-Oriented Programming in Java **CO4:** Develop problem-solving and programming skills using OOP concepts **CO5:** Develop the ability to solve real-world problems through software development in high-level programming language like Java
5.		**OPERATING SYSTEMS**	Upon successful completion of the course, a student will be able to: **CO1:** Know Computer system resources and the role of operating system in resource management with algorithms **CO2:** Understand Operating System Architectural design and its services. **CO3:** Gain knowledge of various types of operating systems including Unix and Android. **CO4:** Understand various process management concepts including scheduling, synchronization, and deadlocks. **CO5:** Have a basic knowledge about multithreading. **CO6:** Comprehenddifferent approaches for memory management. **CO7:** Understand and identify potential threats to operating systems and the security features design to guard against them. **CO8:** Specify objectives of modern operating systems and describe how operating systems have evolved over time. **CO9:** Describe the functions of a contemporary operating system

GOVERNMENT DEGREE COLLEGE
NAGARI
Department of Computer Science
COURSE OUTCOMES (2016)

1.	**3-1-108R**	**Computer Fundamentals and Programming in C**	After Completion of this course the student would be able to: **CO1:** Understanding the concept of input and output devices of Computers and how it works and recognize the basic terminology used in computer programming. **CO2:** Write, compile and debug programs in C language and use different data types for writing the programs. **CO3:** Design programs connecting decision structures, loops and functions. **CO4:** Explain the difference between call by value and call by address. **CO5:** Understand the dynamic behavior of memory by the use of pointers. **CO6:** Use different data structures and create / manipulate basic data files and developing applications for real world problems.
2.	**3-2-109**	**Object Oriented Programming Using C++**	After Completion of this course the student would be able to **CO1:** Describe the procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects. **CO2:** Understand dynamic memory management techniques using pointers, constructors, destructors, etc **CO3:** Describe the concept of function overloading, operator overloading, virtual functions and polymorphism. **CO4:** Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming. **CO5:** Demonstrate the use of various OOPs concepts with the help of programs.
3.	**3-3-108**	**Object Oriented Programming Using Java**	After Completion of this course the student would be able to: **CO1:** Understand the use of OOPs concepts. **CO2:** Apply OOPs concepts to solve real world problems **CO3:** Understand the use of abstraction, Packages and Interface in java. **CO4:** Develop Programs for exception handling, multithreaded applications with synchronization. **CO5:** Able to design GUI based applications and develops applets for web applications.

4.	**3-4-108**	**Data Structures**	After Completion of this course the student would be able to: **CO1:** Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms. **CO2:** Apply for arrays, records, linked structures, stacks, queues, trees, and graphs in developing applications. **CO3:** Compare alternative implementations of data structures with respect to performance and benefits of dynamic and static data structures implementations. **CO4:** understand the concept of recursion, and describe how it can be implemented using a stack. **CO5:** calculate computational efficiency of the principal algorithms for sorting, searching, and hashing.
5.	**3-5-111**	**Database Management Systems**	After completing this course satisfactorily, a student will be able to: **CO1:** Understand the fundamental elements of relational database management systems. **CO2:** Gain knowledge concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL. **CO3:** Design ER-models to represent simple database application scenarios. **CO4:** Convert the ER-model to relational tables, populate relational database and formulate SQL queries on data. **CO5:** apply normalization in database design.
6.	**3-5-112**	**Software Engineering**	After completing this course satisfactorily, a student will be able to: **CO1:** Gain knowledge on software engineering principles and techniques. **CO2:** Develop, maintain and evaluate large-scale software systems. **CO3:** Produce efficient, reliable, robust and cost-effective software solutions. **CO4:** Ability to work as an effective member or leader of software engineering teams. **C5:** Understand and meet ethical standards and legal responsibilities.
7.	**3-6-107B**	**Computer Networks**	After completing this course satisfactorily, a student will be able to: **CO1:** Independently understand basic computer network technology. **CO2:** Identify the different types of network topologies and protocols. **CO3:** Explain the types of transmission media with real time applications **CO4:** Gain knowledge on the functions of all layers and their protocols.

			CO5: Understand the routing protocols and analyze how to assign the IP addresses for the given network.
8.	**3-6-107B2**	**Cloud Computing**	After completing this course satisfactorily, a student will be able to: **CO1:** Compare the strengths and limitations of cloud computing. **CO2:** Identify the architecture, infrastructure and delivery models of cloud computing. **CO3:** Apply suitable virtualization concepts. **CO4:** Choose the appropriate cloud player, Programming Models and approach. **CO5:** Address the core issues of cloud computing such as security, privacy and interoperability.
9.	**3-6-107B1**	**Distributed Systems**	After completing this course satisfactorily, a student will be able to: **CO1:** Demonstrate knowledge of the basic elements and concepts related to distributed system technologies and architectural aspects of distributed systems; **CO2:** Understand various distributed algorithms, such as logical clocks and leader election. **CO3:** Design and implement distributed applications; **CO4:** Demonstrate knowledge of details the main underlying components of distributed systems (such as RPC, file systems); **CO5:** Use and apply important methods in distributed systems to support Task Assignment, Load balancing, Migration and threads.