University of Engineering & Management, Jaipur

Syllabus for BSc Computer Science Admission Batch 2026

SEMESTER 1

S. No.	Subject Code	Subject Name	L	T	P	Total Contact Hrs	Credit Points
Theory							
1	BSCCS101	Programming Fundamentals	3	1	0	4	4
2	BSCCS102	Computer System Architecture	3	1	0	4	4
3	BSCCS103	General Studies & Current Affairs - I	3	1	0	4	4
4	BSCCS104	Indian Knowledge System	0	2	0	2	2
Practical							
5	BSCCS191	Programming Fundamentals Lab	0	0	2	2	2
6	BSCCS192	Computer System Architecture Lab	0	0	2	2	2
Sessional							
7	BSCCS193	Soft Skills	0	2	0	2	2
		Total	9	7	4	20	20

Subject Code: BSCCS101

Subject Name: Programming Fundamentals

Credit Points: 04

Course Objective

This course is designed to develop structured as well as object-oriented programming skills using C++ programming language. The course not only focuses on basic C++ constructs but also covers object-oriented programming features in-depth, namely Encapsulation, Abstraction, Inheritance and Polymorphism for writing efficient codes.

Course Learning Outcomes

On successful completion of the course, the student will:

1. Explain significance of object oriented paradigm.
2. Solve programming problems using object oriented features.
3. Reuse classes to create new classes.
4. Handle exceptions in programs.

Detailed Syllabus

Unit 1

Introduction to C++: Overview of Procedural Programming and Object-Oriented Programming, Using main () function, Header Files, Compiling and Executing Simple Programs in C++.

Unit 2

Programming Fundamentals: Data types, Variables, Operators, Expressions, Arrays, Keywords, Naming Convention, Decision making constructs (if, switch), Looping (for, while, do…while), Type Casting, Input-output statements, Functions, Command Line Arguments/Parameters.

Unit 3

Object Oriented Programming: Overview of Abstraction, Encapsulation, Inheritance, and Polymorphism. Creating Classes and objects, Modifiers and Access Control, Constructors, Implementation of Inheritance (Single and multilevel), Implementation of Polymorphism (Function Overloading and Operator Overloading, Function Overriding).

Unit 4

Pointers and References: Static and dynamic memory allocation, Pointer and Reference Variables, Pointers vs. References, Implementing Runtime polymorphism using pointers and references.

Unit 5

Exception and File Handling: Using try, catch, throw, throws and finally; Nested try, creating user defined exceptions, File I/O Basics, File Operations.

References

1. Forouzan & Gilbert (2012). Computer Science: A Structured Approach Using C++. Cengage Learning.

2. Schildt, H. (2003). C++: The Complete Reference. 4th edition. Tata McGraw-Hill.

Additional Resources:

1. Balaguruswamy, E. (2017). Object Oriented Programming with C++ (7th ed.). McGraw Hill Education.

2. Kanetkar, Y. P. (2015). Let us C++ .2nd edition. BPB Publishers.

3. Prata, S. (2015). C++ Primer Plus 6th edition. Pearson Education India.

4. Stroustrup, B. (2013). The C++ Programming Language .4th Edition. Pearson Education.

Subject Code: BSCCS191

Subject Name: Programming Fundamentals Lab

Credit Points: 02

Practical

1. Write a program to compute the sum of the first n terms of the following series: S = 1 - 1 / (2 ^ 2) + 1 / (3 ^ 3) - ... 1 / (n ^ n) where ^ is exponentiation. The number of terms n is to be taken from user through command line. If command line argument is not found then prompt the user to enter the value of n.
2. Write a program to remove the duplicates from an array.
3. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
4. Write a menu driven program to perform following operations on strings (without using inbuilt string functions):
   a) Show address of each character in string
   b) Concatenate two strings.
   c) Compare two strings
   d) Calculate length of the string (use pointers)
   e) Convert all lowercase characters to uppercase

f) Reverse the string

5. Write a program to merge two ordered arrays to get a single ordered array.

6. Write a program to search a given element in a set of N numbers using Binary search (i) with recursion (ii) without recursion.

7. Write a program to calculate GCD of two numbers (i) with recursion (ii) without recursion.

8. Create Matrix class. Write a menu-driven program to perform following Matrix operations

a) Sum

b) Product

c) Transpose

9. Define a class Person having name as a data member. Inherit two classes Student and Employee from Person. Student has additional attributes as course, marks and year and Employee has department and salary. Write display() method in all the three classes to display the corresponding attributes. Provide the necessary methods to show runtime polymorphism.

10. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.

11. Write a program to read two numbers p and q. If q is 0 then throw an exception else display the result of p/q.

12. Rewrite Matrix class of Q8 with exception handling. Exceptions should be thrown by the functions if matrices passed to them are incompatible and handled by main() function.

13. Create a class Student containing fields for Roll No., Name, Class, Year and Total Marks. Write a program to store 5 objects of Student class in a file. Retrieve these records from file and display them.

14. Copy the contents of one text file to another file, after removing all whitespaces.

Subject Code: BSCCS102

Subject Name: Computer System Architecture

Credit Points: 04

Course Objective

This course introduces the students to the fundamental concepts of digital computer organization, design and architecture. It aims to develop a basic understanding of the building blocks of the computer system and highlights how these blocks are organized together to architect a digital computer system.

Course Learning Outcomes

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

1. Design Combinational Circuits using basic building blocks. Simplify these circuits using

Boolean algebra and Karnaugh maps. Differentiate between combinational circuits and

sequential circuits.

2. Represent data in binary form, convert numeric data between different number systems and

perform arithmetic operations in binary.

3. Determine various stages of instruction cycle and describe interrupts and their handling.

4. Explain how CPU communicates with memory and I/O devices.

5. Simulate the design of a basic computer using a software tool

Detailed Syllabus

Unit 1

Digital Logic Circuits: Logic Gates, truth tables, Boolean Algebra, digital circuits, combinational circuits, sequential circuits, circuit simplification using Karnaugh map, Don't Care Conditions, flip-flops, characteristic tables

Unit 2

Digital Components: Half Adder, Full Adder, Decoders, Multiplexers, Registers and Memory Units

Unit 3

Data Representation and Basic Computer Arithmetic: Number system, complements, fixed and floating point representation. Alphanumeric representation. Addition, subtraction.

Unit 4

Basic Computer Organization and Design: Common Bus system, instruction codes, instruction format, instruction set completeness, Sequence Counter, timing and control, instruction cycle, memory reference instructions and their implementation using arithmetic, logical, program control, transfer and input output micro-operations, interrupt cycle.

Unit 5

Central Processing Unit: Micro programmed Control vs Hardwired Control, lower level programming languages, Instruction format, accumulator, general register organization, stack organization, zero-address instructions, one-address instructions, two-address instructions, three-address instructions, Addressing Modes, RISC, CISC architectures, pipelining and parallel processing.

Unit 6

Memory Organization and Input-Output Organization: Input-Output Organization: Peripheral Devices, I/O interface, I/O vs. Memory Bus, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access

References

1. Mano, M. (1992). Computer System Architecture. 3rd edition. Pearson Education.

Additional Resources

1. Mano, M. (1995). Digital Design. Pearson Education Asia.

2. Null, L., & Lobur, J. (2018). The Essentials of Computer Organization and Architecture. 5th edition. (Reprint) Jones and Bartlett Learning.

3. Stallings, W. (2010). Computer Organization and Architecture Designing for Performance 8th edition. Prentice Hall of India.

Subject Code: BSCCS192

Subject Name: Computer System Architecture Lab

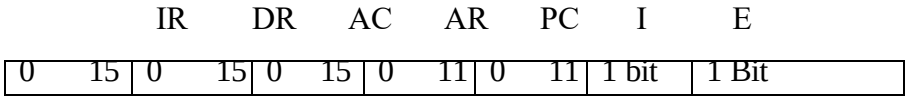
Credit Points: 02

Practical

(Use Simulator – CPU Sim 3.6.9 or any higher version for the implementation)

1. Create a machine based on the following architecture :

Registers



Memory 4096 words 8 bits per word		Instruction format 0 3 4 15 Opcode | Address |

Basic Computer Instructions

Memory Reference			Register Reference		
Symbol	Hex		Symbol	Hex	
AND	0xxx		CLA	E800	
ADD	2xxx		CLE	E400	
LDA	4xxx		CMA	E200	
STA	6xxx		CME	E100	
BUN	8xxx	Direct Addressin	CIR	E080	
			CIL	E040	

ISZ	Cxxx	g	INC	E020	
AND_I	1xxx		SPA	E010	
ADD_I	3xxx		SNA	E008	
LDA_I	5xxx		SZA	E004	
STA_I	7xxx	Indirect	SZE	E002	
BUN_I	9xxx	Addressin	HLT	E001	
ISZ_I	Dxxx	g			

Refer to Chapter-5 of reference 1 for description of instructions.

Design the register set, memory and the instruction set. Use this machine for the assignments of this section.

1. Create a Fetch routine of the instruction cycle.
2. Write an assembly program to simulate ADD operation on two user-entered numbers.
3. Write an assembly program to simulate SUBTRACT operation on two user-entered numbers.
4. Write an assembly program to simulate the following logical operations on two user-entered numbers.
    1. AND
    2. OR
    3. NOT
    4. XOR
    5. NOR
    6. NAND
5. Write an assembly program to simulate MULTIPLY operation on two user-entered numbers.

6. Write an assembly program for simulating following memory-reference instructions.

    1. ADD
    2. LDA
    3. STA
    4. BUN
    5. ISZ

7. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:

1. CLA
2. CMA
3. CME
4. HLT

8. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:

1. INC
2. SPA
3. SNA
4. SZE

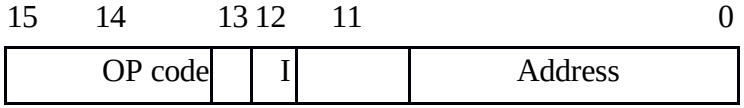
9. Write an assembly language program to simulate the machine for following register reference instructions and determine the

contents of AC, E, PC, AR and IR registers in decimal after the execution:

1. CIR
2. CIL

10. Write an assembly program that reads in integers and adds them together; until a negative non-zero number is read in. Then it outputs the sum (not including the last number).
11. Write an assembly program that reads in integers and adds them together; until zero is read in. Then it outputs the sum.
12. Create a machine for the following instruction format:

Instruction format



The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit address.

Write an assembly program to simulate the machine for addition of two numbers with I= 0 (Direct Address) and address part = 082. The instruction to be stored at address 022 in RAM, initialize the memory word with any decimal value at address 082. Determine the contents of AC, DR, PC, AR and IR in decimal after the execution.

13. Simulate the machine for the memory-reference instruction referred in above question with I= 1 (Indirect Address) and address part = 082. The instruction to be stored at address 026 in RAM. Initialize the memory word at address 082 with the value 298. Initialize the memory word at address 298 with operand 632 and AC with 937. Determine the contents of AC, DR, PC, AR and IR in decimal after the execution.
14. The instruction format contains 3 bits of opcode, 12 bits for address and 1 bit for addressing mode. There are only two addressing modes, I = 0 is direct addressing and I = 1 is indirect addressing. Write an assembly program to check the I bit to

determine the addressing mode and then jump accordingly.