B.A. /B.Sc. (Computer Applications)

CBCS Pattern in Semester System - 2019

Semester -I			
Course Title	**H/Week**		**Credits**
Programming in C	Th 4	Pr 3	4+1 = 5
Semester –II			
Programming in C++	4	3	4+1 = 5
Semester –III			
Relational Data Base Management Systems	4	3	4+1 = 5
Semester –IV			
Multi Media Systems	4	3	4+1 = 5
Semester –V			
Mobile Applications	4	3	4+1 = 5
Semester –VI			
Web Technologies	4	3	4+1 = 5

AECC

Semester -I	Hours/Week	Credits
Fundamentals of Computer	Th 2	2
Semester -II	Hour/Week	
Office Automation	2	2
SEC		
Semester -III		
Python –I (Sec –I)	2	2
Ci Lab –I (Sec –II)	2	2
Semester -IV		
Python –II (Sec –III)	2	2
Ci Lab –II (Sec –IV)	2	2
Generic Elective (GE)		
Semester -IV		
Information Technologies	4	4
Project/Optional		
Information Security and	Thr pr	3+1=4

Cyber Laws	3	3	

Programming in C

Semester -I

Theory	4 Hours/Week	4 credits
Practical	3 Hours/Week	1 credit

Unit – I

Computer Fundamentals: Introduction of Computers, Classification of Computers, Anatomy of a Computer, Memory Hierarchy, Introduction to OS, Operational Overview of a CPU.
Program Fundamentals: Generation and Classification of Programming Languages, Compiling, Interpreting, Loading, Linking of a Program, Developing Program, Software Development.
Algorithms: Definitions, Different Ways of Stating Algorithms (Step-form, Pseudo-code, Flowchart), Strategy for Designing Algorithms, Structured Programming Concept.
Basics of C: Overview of C, Developing Programs in C, Parts of Simple C Program, Structure of a C Program, Comments, Program Statements, C Tokens, Keywords, Identifiers, Data Types, Variables, Constants, Operators and Expressions, Expression Evaluation–precedence and associativity, Type Conversions.

Unit – II

Input-Output: Non-formatted and Formatted Input and Output Functions, Escape Sequences,
Control Statements: Selection Statements – if, if-else, nested if, nested if-else, comma operator, conditional operator, switch; Iterative Statements–while, for, do-while; Special Control Statement–goto, break, continue, return, exit.
Arrays and Strings: One and Two Dimensional Arrays, Character Arrays, Functions from ctype.h, string.h.

Unit – III

Functions: Concept of Function, Using Functions, Call-by-Value Vs Call-by-reference, Passing Arrays to Functions, Score of Variables, Storage Classes, Inline Functions, and Recursion.
Pointers: Introduction, Address of Operator (&), Pointer, Uses of Pointers, Arrays and Pointers, Pointers and Strings, Dynamic Memory Allocation.

Unit – IV

User-Defined Data Types: Declaring a Structure (Union) and its members, Initialization Structure (Union), Accessing members of a Structure (Union), Structures verses Unions, Enumeration Types.
Files: Introduction, Using Files, Working with Text Files and Binary Files, Other File Management Functions.

Text Pradip Dey, Manas Ghosh, Computer Fundamentals and Programming in C (2e)

References

1. **Ivor Horton, Beginning C**
2. **Herbert Schildt, The Complete Reference C**
3. **Paul Deitel, Harvey Deitel, C How To Program**
4. **Byron S. Gottfried, Theory and Problems of Programming with C**
5. **Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language**
6. **B. A. Forouzan, R. F. Gilberg, A Structured Programming Approach Using C**

With effect from the Academic Year 2019-2020

With Effect from the Academic Year 2019-2020

C Lab (Semester –I)

Credit :1

1.Write a program to find the largest two numbers using if and conditional operator.

2.Write a program to calculate arithmetic operations of two numbers using switch.

3.Write a program to print the reverse of a given number.

4. Write a program to print whether the given number is a prime or not.

5. Write a program to find largest and smallest elements in a given list of numbers.

6. Write a program to find the sum of two matrices.

7 Write a program to find the product of two matrices.

8 Write a program to print the reverse of a given string.

9 Write a program to find the factorial of a positive integer using iteration and recursion.

10 Write a program to find the GCD of two positive integers using iteration and recursion.

11 Write a program to demonstrate the call by value and the call by reference concepts.

12 Write a program to illustrate the use of Enumeration data type.

13 Write a program to illustrate the use of structure concept.

14 Write a program to illustrate the use of union concept.

15 Write a program to write content into a file and display contents of a file

16 Write a program to copy content of one file into another file and display the content of new file.

Note : Write the Pseudo Code and draw Flow Chart for the above programs.

Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows 10.

Programming in C++ (Semester-II

Theory	4 Hours/Week	4 credits
Practical	3 Hours/Week	1 credit

Unit – I

Introduction to C++: Applications, Example Programs, Tokens, Data Types, Operators, Expressions, Control Structures, Arrays, Strings, Pointers, Searching and Sorting Arrays.
Functions: Introduction, Prototype, Passing Data by Value, Reference Variables, Using Reference Variables as Parameters, Inline Functions, Default Arguments, Overloading Functions, Passing Arrays to Functions.

Unit – II

Object Oriented Programming: Procedural Programming verses Object-Oriented Programming, Terminology, Benefits, OOP Languages, and OOP Applications.
Classes: Introduction, Defining an Instance of a Class, Why Have Private Members? Separating Class Specification from Implementation, Inline Member Functions, Constructors, Passing Arguments to Constructors, Destructors, Overloading Constructors, Private Member Functions, Arrays of Objects, Instance and Static Members, Friends of Classes, Member-wise Assignment, Copy Constructors, Operator Overloading.

Unit – III

With Effect from

Inheritance: Introduction, Protected Members and Class Access, Base Class Access Specification, Constructors and Destructors in Base and Derived Classes, Redefining Base Class Functions, Polymorphism and Virtual Member Functions, Abstract Base Classes and Pure Virtual Functions, Multiple Inheritance.
C++ Streams: Stream Classes, Unformatted I/O Operations, Formatted I/O Operations.

Unit – IV

Exceptions: Introduction, Throwing an Exception, Handling an Exception, Object-Oriented Exception Handling with Classes, Multiple Exceptions, Extracting Data from the Exception Class, Re-throwing an Exception.
Templates: Function Templates–Introduction, Function Templates with Multiple Type, Overloading with Function Templates, Class Templates – Introduction, Defining Objects of the Class Template, Class Templates and Inheritance.

Text Tony Gaddis, Starting out with C++: from control structures through objects (7e)

the Academic Year 2019-2020

References

B. Lippman, C++ Primer
Bruce Eckel, Thinking in C++
K.R. Venugopal, Mastering C++
Herbert Schildt, C++: The Complete Reference
Bjarne Stroustrup, The C++ Programming Language
Sourav Sahay, Object Oriented Programming with C++

With the Effect from the Academic Year 2019-2020

C++ Lab (Semester –II)

Practical 3Hours/Week 1 credit

1. Write a program to print the sum of digits of a given number
2. Write a program to check whether the given number is Armstrong or not
3. Write a program to check whether the given string is Palindrome or not
4. Write a program to read student name, roll no, marks and display the same using class and object.
5. Write a program to find area of a rectangle, circle, and square using class and object.
6. Write a program to implement inline function inside and outside of a class for
   a. Finding the area of a square
   b. Finding the area of a cube
7. Write a program to implement friend function and friend class
8. Write a program to implement constructor and destructor with in a class.
9. Write a program to demonstrate hierarchical inheritance.
10. Write a program to demonstrate multiple inheritances.
11. Write a program to demonstrate the constructor overloading.
12. Write a program to demonstrate static polymorphism.
13. Write a program to demonstrate dynamic polymorphism.
14. Write a program to implement polymorphism using pure virtual functions.
15. Write a program to demonstrate the function templates and class templates.
16. Write a program to demonstrate exception handling using try, catch, and finally.

Note :

Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows.

B.A./B.Sc. (Computer Applications)

Model Question Paper Credits -4

3 Hours Max Marks -80

PART -A

Answer any eight questions in part –A **8X4 M = 32 Marks**

UNIT- I 1

2

3

UNIT- II 4

5

6

UNIT- III 7

8

9

UNIT- IV 10

11

12

Part – B **Answer all Questions 12MX4 = 48 Marks**

UNIT- I 13

Or

14

UNIT- II 15

Or

16

UNIT- III 17

Or

18

UNIT- IV 19

Or

20

B.A./B.Sc. (Computer Applications)

Practical Question Paper | Credits -I

3 Hours | Max Marks -50

Answer any Two **15MX2 = 30 MARKS**

UNIT – I	1 Program
UNIT- II	1 Program
UNIT-III	1 Program
UNIT -IV	1 Program

Viva - 10 Marks

Record – 10 Marks

B.Sc. (Computer Science)

CBCS Pattern in Semester System - 2019

Semester -I			
Course Title	**H/Week**		**Credits**
Programming in C	Th 4	Pr 3	4+1 = 5
Semester –II			
Programming in C++	4	3	4+1 = 5
Semester –III			
Data Structures using C++	4	3	4+1 = 5
Semester –IV			
Data Base Management Systems (DBMS)	4	3	4+1 = 5
Semester –V			
Programme in Java	4	3	4+1 = 5
Semester –VI			
Web Technologies	4	3	4+1 = 5

AECC

Semester -I	Hours/Week	Credits
Fundamentals of Computer	Th 2	2
Semester -II	Hour/Week	
Office Automation	2	2
SEC		
Semester -III		
Python –I (Sec –I)	2	2
Operating Systems (Sec –II)	2	2
Semester -IV		
Python –II (Sec –III)	2	2
Operating Systems (Sec –IV)	2	2
Generic Elective (GE)		
Semester -IV		
Information Technologies	4	4
Project/Optional		
Semester -VI		
PHP with MY SQL	Thr pr 3 3	3+1=4

With Effect from the Academic Year 2019-2020

Programming in C Semester -I

Theory	4 Hours/Week	4 credit
Practical	3 Hours/Week	1 credit

Unit – I

Computer Fundamentals: Introduction of Computers, Classification of Computers, Anatomy of a Computer, Memory Hierarchy, Introduction to OS, Operational Overview of a CPU.
Program Fundamentals: Generation and Classification of Programming Languages, Compiling, Interpreting, Loading, Linking of a Program, Developing Program, Software Development.
Algorithms: Definitions, Different Ways of Stating Algorithms (Step-form, Pseudo-code, Flowchart), Strategy for Designing Algorithms, Structured Programming Concept.
Basics of C: Overview of C, Developing Programs in C, Parts of Simple C Program, Structure of a C Program, Comments, Program Statements, C Tokens, Keywords, Identifiers, Data Types, Variables, Constants, Operators and Expressions, Expression Evaluation–precedence and associativity, Type Conversions.

Unit – II

Input-Output: Non-formatted and Formatted Input and Output Functions, Escape Sequences,
Control Statements: Selection Statements – if, if-else, nested if, nested if-else, comma operator, conditional operator, switch; Iterative Statements–while, for, do-while; Special Control Statement–goto, break, continue, return, exit.
Arrays and Strings: One-dimensional Arrays, Character Arrays, Functions from ctype.h, string.h, Multidimensional Arrays.

Unit – III

Functions: Concept of Function, Using Functions, Call-by-Value Vs Call-by-reference, Passing Arrays to Functions, Score of Variables, Storage Classes, Inline Functions, and Recursion.
Pointers: Introduction, Address of Operator (&), Pointer, Uses of Pointers, Arrays and Pointers, Pointers and Strings, Pointers to Pointers, Array of Pointers, Pointer to Array, Dynamic Memory Allocation.

Unit – IV

User-defined Data Types: Declaring a Structure (Union) and its members, Initialization Structure (Union), Accessing members of a Structure (Union), Array of Structures (Union), Structures verses Unions, Enumeration Types.
Files: Introduction, Using Files in C, Working with Text Files, Working with Binary Files, Files of Records, Random Access to Files of Records, Other File Management Functions.

Text

Pradip Dey, Manas Ghosh, Computer Fundamentals and Programming in C (2e)

References BOOKS

Ivor Horton, Beginning C
Ashok Kamthane, Programming in C
Herbert Schildt, The Complete Reference C
Paul Deitel, Harvey Deitel, C How To Program
Byron S. Gottfried, Theory and Problems of Programming with C
Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language
B. A. Forouzan, R. F. Gilberg, A Structured Programming Approach Using C

With Effect from the Academic Year 2019-2020

C Lab Semester -I

Practical 3 Hours/Week 1 credit

1. Write a program to find the largest two (three) numbers using if and conditional operator.
2. Write a program to print the reverse of a given number.
3. Write a program to print the prime number from 2 to n where n is given by user.
4. Write a program to find the roots of a quadratic equation using switch statement.
5. Write a program to print a triangle of stars as follows (take number of lines from user):

```
    *
   ***
  *****
 *******
*********
```

6. Write a program to find largest and smallest elements in a given list of numbers.
7. Write a program to find the product of two matrices..
8. Write a program to find the GCD of two numbers using iteration and recursion.
9. Write a program to illustrate use of storage classes.
10. Write a program to demonstrate the call by value and the call by reference concepts.
11. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program to illustrate use of data type enum.
13. Write a program to demonstrate use of string functions string.h header file.
14. Write a program that opens a file and counts the number of characters in a file.
15. Write a program to create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
16. Write a program that opens an existing text file and copies it to a new text file with all lowercase letters changed to capital letters and all other characters unchanged.

Note : Write the Pseudo Code and draw Flow Chart for the above programs.
Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows 10.

With Effect from the Academic Year 2019-2020

Programming in C++ Semester -II

Theory	4 Hours/Week	4 credits
Practical	3 Hours/Week	1 credit

Unit – I

Introduction to C++: Applications, Example Programs, Tokens, Data Types, Operators, Expressions, Control Structures, Arrays, Strings, Pointers, Searching and Sorting Arrays.
Functions: Introduction, Prototype, Passing Data by Value, Reference Variables, Using Reference Variables as Parameters, Inline Functions, Default Arguments, Overloading Functions, Passing Arrays to Functions.
Object Oriented Programming: Procedural and Object-Oriented Programming, Terminology, Benefits, OOP Languages, and OOP Applications.

Unit – II

Classes: Introduction, Defining an Instance of a Class, Why Have Private Members? Separating Class Specification from Implementation, Inline Member Functions, Constructors, Passing Arguments to Constructors, Destructors, Overloading Constructors, Private Member Functions, Arrays of Objects, Instance and Static Members, Friends of Classes, Member-wise Assignment, Copy Constructors, Operator Overloading, Object Conversion, Aggregation.

Unit – III

Inheritance: Introduction, Protected Members and Class Access, Base Class Access Specification, Constructors and Destructors in Base and Derived Classes, Redefining Base Class Functions, Class Hierarchies, Polymorphism and Virtual Member Functions, Abstract Base Classes and Pure Virtual Functions, Multiple Inheritance.
C++ Streams: Stream Classes, Unformatted I/O Operations, Formatted I/O Operations.

Unit – IV

Exceptions: Introduction, Throwing an Exception, Handling an Exception, Object-Oriented Exception Handling with Classes, Multiple Exceptions, Extracting Data from the Exception Class, Re-throwing an Exception, Handling the bad_alloc Exception.
Templates: Function Templates–Introduction, Function Templates with Multiple Type, Overloading with Function Templates, Class Templates – Introduction, Defining Objects of the Class Template, Class Templates and Inheritance, Introduction to the STL.

Text

Tony Gaddis, Starting out with C++: from control structures through objects (7e)

References

B. Lippman, C++ Primer
Bruce Eckel, Thinking in C++
K.R. Venugopal, Mastering C++
Herbert Schildt, C++: The Complete Reference
Bjarne Stroustrup, The C++ Programming Language
Sourav Sahay, Object Oriented Programming with C++

With Effect from the Academic Year 2019–2020

C++ Lab Semester -II

Practical 3 Hours/Week 1 credit

1 Write a program to.

a. Print the sum of digits of a given number.

b. Check whether the given number is Armstrong or not

c. Print the prime number from 2 to n where n is natural number given.

2 Write a program to find largest and smallest elements in a given list of numbers and sort the given list.

3 Write a program to read the student name, roll no, marks and display the same using class and object.

4 Write a program to implement the dynamic memory allocation and de-allocation using new and delete operators using class and object.

Write a program to find area of a rectangle, circle, and square using constructors.

5

6 Write a program to implement copy constructor.

7 Write a program using friend functions and friend class.

8 Write a program to implement constructors
§ Default Constructor, Parameterized Constructor, Copy Constructor
§ Define the constructor inside/outside of the class
§ Implement all three constructors within a single class as well as use multiple classes(individual classes)

Write a program to implement the following concepts using class and object
§ Function overloading
§ Operator overloading (unary/binary(+ and -))

Write a program to demonstrate single inheritance, multilevel inheritance and multiple inheritances.

Write a program to implement the overloaded constructors in inheritance.

Write a program to implement the polymorphism and the following concepts using class and object.
§ Virtual functions
§ Pure virtual functions

Write a program to implement the virtual concepts for following concepts
§ Constructor (not applied)
§ Destructor (applied)

Write a program to demonstrate static polymorphism using method overloading.

Write a program to demonstrate dynamic polymorphism using method overriding and dynamic method dispatch.

Write a program to implement the template (generic) concepts
§ Without template class and object
§ With template class and object

Write the Pseudo Code and draw Flow Chart for the above programs.

Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows.

B.Sc. (Computer Science)

Model Question Paper — Credits -4

3 Hours — Max Marks -80

PART -A

Answer any eight questions in part –A **8X4 M = 32 Marks**

UNIT- I 1

2

3

UNIT- II 4

5

6

UNIT- III 7

8

9

UNIT- IV 10

11

12

Part – B **Answer all Questions 12MX4 = 48 Marks**

UNIT- I 13

Or

14

UNIT- II 15

Or

16

UNIT- III 17

Or

18

UNIT- IV 19

Or

20

B.Sc. (Computer Science)

Practical Question Paper | Credits -I

3 Hours | Max Marks -50

Answer any Two | 15MX2 = 30 MARKS

UNIT – I	1 Program
UNIT- II	1 Program
UNIT-III	1 Program
UNIT -IV	1 Program

Viva - 10 Marks

Record – 10 Marks