

Credit Based Choice Based Curriculum for

Bachelor of Computer Applications (BCA) Programme 2016 Onwards

| S. No. | Subject Code | Name of the Subject | Credit Details | Credit Hours | | |
|------------------------------------|--------------|---------------------------------------|----------------|---------------|---|---|
| | | | | L | T | P |
| Core Courses | | | | | | |
| 1. | BCA-16101CC | Programming in C/C++ | 4 | 4 | 0 | 0 |
| | | Lab: Programming in C/C++ | 2 | 0 | 0 | 4 |
| 2. | BCA-16102CC | Computer System Architecture | 4 | 4 | 0 | 0 |
| | | Lab: Computer System Architecture | 2 | 0 | 0 | 4 |
| Ability Enhancement Courses | | | | | | |
| 3. | BCA-16103AE | English Communication | 2 | 2 | 0 | 0 |
| Generic Elective Courses | | | | | | |
| 4. | BCA-16104GE | Generic Elective 1 (GE1) | 4/5 | 4/5 | | |
| | | Generic Elective 1 Practical/Tutorial | 2/1 | | | |
| Total Credits | | | 20 | Hours = 26/24 | | |

1. Generic Electives:

All the four papers of Generic Electives (GE1 to GE4) should be taken from any ONE of the following disciplines:

- Mathematics
- Statistics
- Operational Research
- Physics
- Electronics
- Commerce
- Economics

2. Skill Enhancement Courses:

Currently the following Skill Enhancement courses are offered. If required, these elective courses will be modified/changed during next BOS meeting based on market demand.

- Android Programming (1) + Lab (2)
- PHP Programming (1) + Lab (2)

3. Discipline Specific Electives:

Currently the following Discipline Specific Elective courses are offered. If required, these elective courses will be modified/changed during next BOS meeting based on market demand.

- Numerical Methods (4) + Lab(4)
- Data Mining (4) + Lab(4)
- Cloud Computing (4) + Lab(4)
- Dissertation/Project (4) + Lab(4)

4. Generic Electives (GE) (Minor – Computer Science) for non-Computing Disciplines:

Currently the following Generic Elective courses in Computing are offered for those students pursuing degree in a discipline other than Computing. If required, these courses will be modified/changed during next BOS meeting based on market demand.

- Computer Fundamentals (4) + Lab (4)
- Introduction to Database Systems (4) + Lab (4)
- Introduction to Programming (4) + Lab (4)
- Computer Networks and Internet Technologies (4) + Lab (4)

Core Courses:

BCA-16101CC Programming Fundamentals using C/C++

Theory: 60 Lectures

1. Introduction to C and C++ (3 Lectures)

History of C and C++, Overview of Procedural Programming and Object-Oriented Programming, Using main() function, Compiling and Executing Simple Programs in C++.

2. Data Types, Variables, Constants, Operators and Basic I/O (5 Lectures) Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putchar), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.h etc).

3. Expressions, Conditional Statements and Iterative Statements (5 Lectures) Simple Expressions in C++ (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct), Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)

4. Functions and Arrays (10 Lectures) Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments. Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Twodimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays

5. Derived Data Types (Structures and Unions) (3 Lectures) Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, Manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.

6. Pointers and References in C++ (7 Lectures) Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with Pointers, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, Using references as function arguments and function return values

7. Memory Allocation in C++ (3 Lectures) Differentiating between static and dynamic memory allocation, use of malloc, calloc and free functions, use of new and delete operators, storage of variables in static and dynamic memory allocation

8 . File I/O, Preprocessor Directives (4 Lectures)

Opening and closing a file (use of fstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros

9. Using Classes in C++ (7 Lectures)

Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables & Functions, Objects as parameters, Specifying the Protected and Private Access, Copy Constructors, Overview of Template classes and their use.

10. Overview of Function Overloading and Operator Overloading (5 Lectures)

Need of Overloading functions and operators, Overloading functions by number and type of arguments, Looking at an operator as a function call, Overloading Operators (including assignment operators, unary operators)

11. Inheritance, Polymorphism and Exception Handling (8 Lectures)

Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions), Basics Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions, Rethrowing exceptions.

Reference Books

1. HerbtzSchildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill.2003 th
2. BjarneStroustrup, "The C++ Programming Language", 4 Edition, Addison-Wesley , 2013.
3. BjarneStroustrup, "Programming -- Principles and Practice using C++", 2nd Edition, Addison-Wesley 2014.
4. E Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Education, 2008.
5. Paul Deitel, Harvey Deitel, "C++ How to Program", 8th Edition, Prentice Hall, 2011.
5. John R. Hubbard, "Programming with C++", Schaum's Series, 2nd Edition, 2000.
6. Andrew Koeni, Barbara, E. Moo, "Accelerated C++", Published by Addison-Wesley , 2000.
7. Scott Meyers, "Effective C++", 3rd Edition, Published by Addison-Wesley, 2005.
8. Harry, H. Chaudhary, "Head First C++ Programming: The Definitive Beginner's Guide", First Create space Inc, O-D Publishing, LLC USA.2014
9. Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.
10. Stanley B. Lippman, JoseeLajoie, Barbara E. Moo, "C++ Primer", Published by AddisonWesley, 5th Edition, 2012

LAB BCA-16101CC : Programming Fundamentals using C/C++

Practical: 60 Lectures

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series $S = 6 + 6/7 + 6/7^2 + 6/7^3 + \dots$
4. WAP to compute the sum of the first n terms of the following series
 $S = 1 - 2 + 3 - 4 + 5 - 6 + \dots$
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. WAP to print a triangle of stars as follows (take number of lines from user):

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

10. WAP to perform following actions on an array entered by the user:
 - i) Print the even-valued elements
 - ii) Print the odd-valued elements
 - iii) Calculate and print the sum and average of the elements of array
 - iv) Print the maximum and minimum element of array
 - v) Remove the duplicates from the array
 - vi) Print the array in reverse order

The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.

15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.
16. Write a menu driven program to perform following operations on strings:
 - a) Show address of each character in string
 - b) Concatenate two strings without using strcat function.
 - c) Concatenate two strings using strcat function.
 - d) Compare two strings
 - e) Calculate length of the string (use pointers)
 - f) Convert all lowercase characters to uppercase
 - g) Convert all uppercase characters to lowercase
 - h) Calculate number of vowels
 - i) Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation):
 - a) Sum b) Difference c) Product d) Transpose
22. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
23. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.
24. Create a class Box containing length, breath and height. Include following methods in it:
 - a) Calculate surface Area
 - b) Calculate Volume
 - c) Increment, Overload ++ operator (both prefix & postfix)
 - d) Decrement, Overload -- operator (both prefix & postfix)
 - e) Overload operator == (to check equality of two boxes), as a friend function
 - f) Overload Assignment operator
 - g) Check if it is a Cube or cuboid

Write a program which takes input from the user for length, breath and height to test the above class.

25. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.

26. Write a program to retrieve the student information from file created in previous question and print it in following format:

Roll No. Name Marks

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

28. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.

29. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

BCA-16102CC: Computer System Architecture

Theory: 60 Lectures

1. Introduction

(8 lectures)

Logic gates, boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexers, registers, counters and memory units.

2 Data Representation and Basic Computer Arithmetic (10 lectures)

Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison, multiplication and division algorithms for integers

3 . Basic Computer Organization and Design (13 lectures)

Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Interconnection Structures, Bus Interconnection design of basic computer.

4 . Central Processing Unit (15 lectures)

Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control. Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming, RISC, CISC architectures, pipelining and parallel architecture.

5. Memory Organization (6 lectures)

Cache memory, Associative memory, mapping.

6. Input-Output Organization (8 lectures)

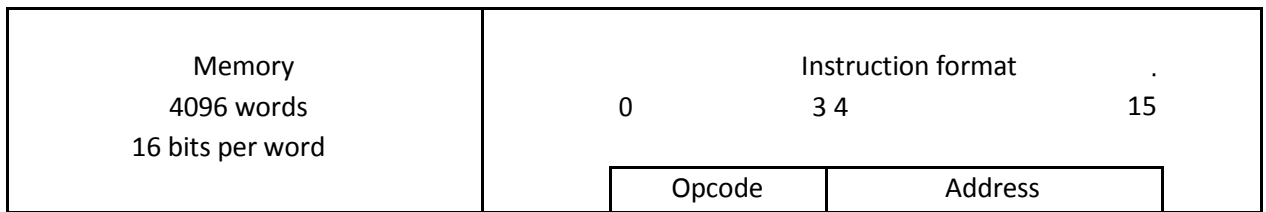
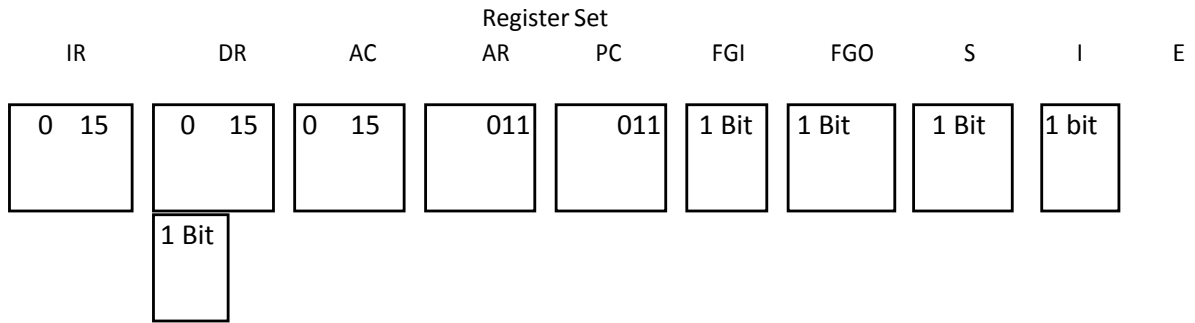
Input / Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels.

Recommended Books:

1. M. Mano, Computer System Architecture, Pearson Education 1992
2. A. J. Dos Reis, Assembly Language and Computer Architecture using C++ and JAVA, Course Technology, 2004
3. W. Stallings, Computer Organization and Architecture Designing for Performance, 8 Edition, Prentice Hall of India, 2009
4. M.M. Mano , Digital Design, Pearson Education Asia, 2013
5. Carl Hamacher, Computer Organization, Fifth edition, McGrawHill, 2012.

LAB BCA-16102CC : Computer System
Architecture Lab Practical: 60 Lectures

1. Create a machine based on the following architecture:



Basic Computer Instructions

| Memory Reference | | Register Reference | | Input-Output | |
|------------------|------|--------------------|------|--------------|------|
| Symbol | Hex | Symbol | Hex | Symbol | Hex |
| AND | 0xxx | CLA | E800 | INP | F800 |
| ADD | 2xxx | CLE | E400 | OUT | F400 |
| LDA | 4xxx | CMA | E200 | SKI | F200 |
| STA | 6xxx | CME | E100 | SKO | F100 |
| BUN | 8xxx | CIR | E080 | ION | F080 |
| BSA | Axxx | CIL | E040 | IOF | F040 |
| ISZ | Cxxx | INC | E020 | | |
| AND_I | 1xxx | SPA | E010 | | |
| ADD_I | 3xxx | SNA | E008 | | |
| LDA_I | 5xxx | SZA | E004 | | |
| STA_I | 7xxx | SZE | E002 | | |
| BUN_I | 9xxx | HLT | E001 | | |
| BSA_I | Bxxx | | | | |
| ISZ_I | Dxxx | | | | |

Refer to Chapter-5 of Morris Mano for description of instructions.

2. Create the micro operations and associate with instructions as given in the chapter (except interrupts). Design the register set, memory and the instruction set. Use this machine for the assignments of this section.
3. Create a Fetch routine of the instruction cycle.
4. Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions:

- | | | |
|--------|--------|--------|
| a. CLA | e. CIR | i. SNA |
| b. CLE | f. CIL | j. SZA |
| c. CMA | g. INC | k. SZE |
| d. CME | h. SPA | l. HLT |

Initialize the contents of AC to (A937)₁₆, that of PC to (022)₁₆ and E to 1.

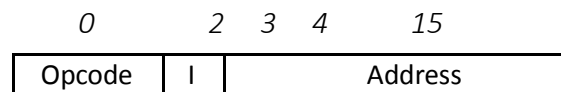
5. Simulate the machine for the following memory-reference instructions with I= 0 and address part = 082. The instruction to be stored at address 022 in RAM. Initialize the memory word at address 082 with the operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

- | | |
|--------|--------|
| a. ADD | f. BSA |
| b. AND | g. ISZ |
| c. LDA | |
| d. STA | |
| e. BUN | |

6. Simulate the machine for the memory-reference instructions referred in above question with I= 1 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 B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

7. Modify the machine created in Practical 1 according to the following instruction format:

Instruction format



- a. The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit address. There are only two addressing modes, I = 0 (direct addressing) and I = 1 (indirect addressing).
- b. Create a new register I of 1 bit.
- c. Create two new microinstructions as follows :
 - i. Check the opcode of instruction to determine type of instruction (Memory Reference/Register Reference/Input-Output) and then jump accordingly.
 - ii. Check the I bit to determine the addressing mode and then jump accordingly.

BCA-16103AE English Communication

Syllabus at the Undergraduate level for AECC/English under Choice Based Credit Scheme (CBCS) to be implemented from 2017.

The detailed syllabus given below has been designed in compliance with the UGC scheme of CBCS and the decisions taken by the University of Kashmir and the suggestions of the Board of Undergraduate Studies (English) respectively.

In Semester I, AECC-I (Ability Enhancement Compulsory Course) will be offered for all the streams. The weightage of this course shall be 2 Credits and it will comprise of 2 Units.

In Semester II, AECC-II will be offered for all the streams. Its weightage will be 2 Credits and it will comprise of 2 Units.

In Semester III, English-I will be offered. This course shall have the weightage of 6 Credits comprising of 4 units. This course will be offered for Arts and Commerce Streams only.

For Science stream, only the first two units of two credits from the course titled English-I will be offered.

In Semester IV, English-II will be offered. This course shall have the weightage of 6 Credits comprising of 4 units. This course will be offered for Arts and Commerce Streams only.

For Science stream, only the first two units of two credits from the course titled English-II will be offered.

In Semesters V and VI, English-III and English-IV will be offered respectively for all the streams with the weightage of 2 Credits each and comprising of 2 Units each.

Effective from academic session-2017

Semester I

Credits-2 (Prescribed for all streams of B.A/B.Sc./B.Com)

AECC-1:

Unit I:

- **Communication: An Overview**

- Listening Skills : Types/Barriers/Traits of a Good Listener

Unit 2:

- **Language in Practice : Interviews / Presentations**

- Language in Practice : Group Discussions/Telephone Skills

Recommended Text Book: *Vibrant English*. Orient BlackSwan

Suggested Reading: *Fluency in English-II*. OUP.