FACULTY OF COMPUTER APPLICATIONS
Bachelor of Computer Applications

- **Sem.** : B.Sc.IT-2
- **Subject Code** : 05BS0202
- **Subject** : Programming Techniques- 2 (C++)
- **Course Objectives** :

1. To make students understand concepts of Class and Objects.
2. To make students understand concepts of Inheritance, Polymorphism.
3. To make students understand the basic concepts of Constructors/Destructors.
4. To make students understand function overloading, operator overloading, virtual functions.
5. To make students understand concepts of arrays, pointers, dynamic memory allocation

- **Prerequisites** :
  1. Basics fundamental knowledge of c language
  2. Programming concepts including algorithms, flow chart, and logic building

Unit No	Topics Covered	No of lectures required
1	**Object oriented programming paradigm:** Object oriented paradigm, structured versus Object-Oriented Development, Elements of Object oriented programming, objects, classes, multiple views of the same object, encapsulation and data abstraction, inheritance, delegation, polymorphism, message communication, merits and demerits of OO methodology.	10
2	**C++ Programming Basics** Getting Started, Basic Program Construction, Output Using cout, Directives, Comments, Integer Variables, Character Variables, Input with cin, Floating Point Types, Type bool, The setw Manipulator, Variable Type Summary, Type Conversion, Arithmetic Operators,	10

	Library Functions **Loops and Decisions, Structures, Enumerations** Relational Operators, Loops, Decisions, Logical Operators, , Precedence Summary, Other Control Statements, Structures, Enumerations **Functions** Simple Functions, Passing Arguments to Functions, Returning Values from Functions, Reference Arguments, Overloaded Functions, Recursion, Inline Functions, Default Arguments, Scope and Storage Class, Returning by Reference	
3	**Objects and Classes** A Simple Class, C++ Objects as Physical Objects, C++ Objects as Data Types, Constructors, Objects as Function Arguments, The Default Copy Constructor, Returning Objects from Functions, Structures and Classes, Classes, Objects, and Memory, Static Class Data, const and Classes **Arrays and Strings** Array Fundamentals, Arrays as Class Member Data, Arrays of Objects, C-Strings, The Standard C++ string Class **Operator Overloading** Overloading Unary Operators, Overloading Binary Operators, Data Conversion, Pitfalls of Operator Overloading and Conversion, Keywords explicit and mutable	**10**
4	**Inheritance** Derived Class and Base Class, Derived Class Constructors, Overriding Member Functions, Which Function Is Used?, Class hierarchies, public and private inheritance, levels of inheritance, multiple inheritance, ambiguity in multiple inheritance, aggregation **Pointers** Addresses and pointers, the address-of operator &, pointers and arrays, pointers and functions, pointers and C-Type strings, memory management: new and delete, pointers to objects **Virtual Functions** Virtual functions, friend functions, static functions, assignment and copy initialization, this pointer, dynamic type information	**10**
5	**Streams and Files** Stream classes, stream errors, Disk file I/O with streams, file pointers, error handling in file I/O, File I/O with member functions, overloading the extraction and insertion operators, memory as stream object, command-line arguments, printer output **Exception Handling** Exceptions,	**8**

	components of exception, handling multiple exceptions	

Course Outcomes :

1. Differentiate the procedural and object-oriented paradigm
2. Demonstrate the use of various OOPs concepts with the help of programs
3. Understand dynamic memory management techniques using pointers, constructors, destructors
4. Describe the concept of function overloading, operator overloading, virtual functions and polymorphism
5. Classify inheritance with the understanding of early and late binding.

Course Outcomes – Program Outcomes Mapping Table :

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1		H							H		
CO2	M	H			H						H
CO3	L	H			H						M
CO4		M			H				L		H
CO5	H	H			H						M

Text Book :

1. "Object Oriented Programming in C++"; Robert Lafore; 4th Edition; Pearson Education
2. "Mastering C++"; K R Venugopal, Rajkumar, T Ravishankar; Tata McGRAW HILL

Reference Books :

1. "Object Oriented programming with C++"; E. Balagurusamy; TMH
2. "Complete Reference C++"; Herbert Schildt; McGraw Hill Publications
3. "Computer Science- A Structured approach using C++"; Forouzan ,Gilburg; THOMSON Books

4. "The C++ Programming Language"; Bjarne Stroustrup; Pearson Education
5. "Effective C++"; Scott Mayer; Addison Wesley

Web References :

https://www.geeksforgeeks.org/c-plus-plus/

App References :

1. **NA**

Syllabus Coverage from text /reference book & web/app reference:

Unit #	Chapter Numbers
1	Book-2 : chapter1,
2	Book-1 :chapter 2,3,4,5
3	Book-1 :chapter 6,7,8
4	Book-1 :chapter 9,10,11
5	Book-1 : chapter 10, book-2 : chapter 19

Marwadi University

FACULTY OF COMPUTER APPLICATIONS
Bachelor of Computer Applications
PRACTICALS

Sr. No.	List of Practical
1	**1.** Write a program that computes and displays the area of a triangle, after the user has entered its height and base. **2.** Write a program that inputs an integer number and displays its absolute value (i.e., if the input value is negative then the value would be displayed as positive and if the input value is positive or 0 then the output should coincide with the input. For example, if the input is –3 the output value will be 3). **3.** Write a program that reads in 10 numbers and displays their average, and the largest of the numbers (assume that all the inputs are positive integers). **4.** Illustrate the concept of declaring a variable in C++ program, as and when it is required not necessarily beginning of the program. **5.** Write a program that reads in 10 numbers and displays their average, and the largest of the numbers (assume that all the inputs are positive integers).
2	**1.** Illustrate the use of reference variable through the C++ program. **2.** Illustrate the concept of dynamic variable in C++. **3.** Demonstrate the use of static variables in a class by using it to count the number of objects created in the program **4.** WAP in C++ to compute and display the salary of employee. Basic=10,000 DA=51% and HRA=15% of basic using the manipulator setw and endl to display the write justify result. **5.** Write a program that prompts the user to enter 5 exam marks (design your program such that it would be easy to change if we wanted it to work for more than 5 exams). The program then displays the average mark, the highest mark, and number of marks entered less than the average mark. For example, when executing the program and inputting 20, 30, 10, 40, and 50 the running session will look as follows: Enter a mark: 20 **6.** Write a program which illustrates the use of scope resolution operator.

Marwadi University

FACULTY OF COMPUTER APPLICATIONS
Bachelor of Computer Applications

<table>
<tr>
<td></td>
<td>
7. Define a class employee with the following data members as private: Empno, ename, basic, hra, da, salary and a private member function Called, float calculate(void) - to calculate net salary.
Public member functions as bellow : Void getinfo(void) – A function to accept values for empno, name, basic, hra, da and invoke calculate( ) function to calculate net salary. Void displayinfo(void) – A function to display all the data members on the screen. Write an appropriate main( ) function for it.
8. Overload the function volume() to compute volume of a cube, cylinder, rectangular box, and sphere.
Write an inline function to find the area of circle , pass radius as parameter
</td>
</tr>
<tr>
<td>3</td>
<td>
1) Define a class of employees. It should contain employee number, name, address, and number of dependents for the employee. It should also contain function to insert and display information about employee. Define an array of 20 employees. Now write a simple for loop to read information about the employee. At the end, display all employees with more than two dependents.

2) Define a class to represent a bank account. Include the following members :
<table>
<tr><th>Data Members</th><th>Member Functions</th></tr>
<tr><td>Name of Depositor</td><td>To assign initial values</td></tr>
<tr><td>Account Number</td><td>To deposit the amount</td></tr>
<tr><td>Type of Account</td><td>To withdraw an amount after checking the balance amount in account</td></tr>
<tr><td></td><td>To display name and balance</td></tr>
<tr><td colspan="2">Write C++ program to handle 10 customers.</td></tr>
</table>
3) Write a menu driven program that can perform the following functions on strings. (Use overloaded operators where possible). (Do not use predefined string class)
1. Compare two strings for equality (== operator)
2. Check whether first string is smaller than the second (<= operator)
3. Copy the string to another
4. Extract a character from the string (Overload [])
5. Reverse the string 6. Concatenate two strings (+
</td>
</tr>
</table>

	operator) 4) Overload subscript operator [] for a array class. 5) Define a singly linked list class, which is a made up objects of node class. Provide addition, deletion of nodes, with operator overloading.
4	1) Define a class Employees. Also define classes of MaleEmp and FemaleEmp inheriting from that. Define classes Officers, Clerks and peons again inheriting from Employee class. Define an array which contains 10 different types of employees. Define a function ReadDetails() in all above classes. All array elements should be able to be accessed in the same routine irrespective of their type. 2) Define Student class having member data rollno, name, m1, m2, m3, per, result member functions getdata(), dispdata(), cal() for inline functions with reference to class. 3) Illustrate the use of this pointer 4) Implement single, multiple, multilevel, hierarchical and hybrid inheritance using specific example. 5) Write a program to declare friend function in two classes. Calculate the sum of integers of both the classes using friend sum() function. 6) Base Class Name →Counter ; data member → count ; member functions → default constructor to set count to 0, inc() Derived Class Name →Countdown; data member → none; member functions → dec() Note : inc() & dec() must return the count values 7) Write a program to declare protected data in base class. Access data of base class declared under protected section using member functions of derived class.
5	1) Define a class Person. Have data members as name of the person, names of parents of the person, gender, age, and

	an array containing list of interests. Provide functions FindFather, FindMother, FindUndle and FindAunty functions, all of which returns the object of person class. Provide access using function pointer for all these functions. The program should take all information about persons from a file Person. The answers to all queries should also be stored in separate files. 2) Write a C++ program that displays a student object read from a file backwards. 3) Use Time class to provide overloaded. Here the time query is also to be recorded in file. Use C text file to store the query. If a calling function provides expression Time1 – Time2, then operator – function should throw an exception if Time2 is a later time then Time1. Before throwing exception, though, the operator – function should close the file. 4) Use an Employee Class to write records of employee to a file. Include a menu that will allow the user to select any of the following features a. Add a new record. b. Modify an existing record. c. Retrieve and display an entire record for a given name. d. Generate a complete list of all names, addresses and telephone numbers. e. End of the computation. 5) Write a program to illustrate the use of string IO with files.