<table>
<tr><th colspan="9">5194 – COMPUTER PROGRAMMING (OOPS) IN C++</th></tr>
<tr><th colspan="3">Teaching Schedule Per Week</th><th colspan="2" rowspan="2">Progressive Assessment</th><th colspan="4">Examination Schedule (Marks)</th></tr>
<tr><th>Lectures</th><th>Practical</th><th>Credits</th><th colspan="2">Theory</th><th>Practical Ex.</th><th>Total</th></tr>
<tr><td>3</td><td>2</td><td>5</td><td>25</td><td>25</td><td>3 Hrs</td><td>100</td><td>50</td><td>200</td></tr>
<tr><td colspan="2">Pre-requisite</td><td>Source</td><td colspan="6"></td></tr>
<tr><td colspan="2">3030</td><td>PPDCA</td><td colspan="6"></td></tr>
</table>

Semester	Theory	Test	Total	TW	PR	Gr Total
	75	25	100	50	50	200

Rationale:- Object Oriented Programming (OOP) is the most dramatic innovation in software development in the last decade. It should be realised that the chief problem with large computer programs is complexity. OOP offers a new and powerful way to cope with complexity. Its goal is clearer, more reliable resulting in more easily manipulated programs. The key features addressed are inheritance, leading to code reusability, operator overriding and polymorphism, which has made OOP an unparalleled rival in programming parlance.

COURSE CONTENTS Hrs Mks

1. OBJECT ORIENTED PROGRAMMING 3 5

The need of Object Oriented Programming: Procedural languages, The Object Oriented approach, Advantages of Object Oriented Programming.

Characteristics of Object Oriented Languages: Objects, Classes, Inheritance, Reusability, New Data Types, Polymorphism and Overloading

2. AN OVERVIEW OF C++ PROGRAMMING 4 8

Basic Program Construction:- Functions, Program Statements, White Space; Output Using *cout:*-Strings Constants.; c) Pre-processor Directives:-The #include Directive, Header Files.; d) Comments:-Comment Syntax, Usage, Alternative Comment Syntax.; e) Integer Variables:-Definition, Declarations and Definition, Variable Names, Assignment Statements, Integer Constants, Output Variations. ; f) Character Variables:-Character Constants, Initialisation, Escape Sequence. ; g) Input Using cin:-

Variables defined at the Point of Use, Cascading <<, Expressions, Precedence.; h) Type float:-Floating - Point Constants, the *const* Qualifier, The *#define* Directive.; i) Manipulators:-the *endl* Manipulators, The *setw* Manipulators, Type *long*, Cascading the Insertion Operators, Multiple Definition, The *IOMANIP>H* Header file. ; j) Variable Type Summary:-unsigned Data Types. ; k) Type Conversion:- Automatic Conversion, Casting. ; l) Arithmetic Operators:- The Remainder Operators, Arithmetic Assignment Operators, Increment Operators. ; m) Library Functions:- Header Files, Library Files, Header files and Library files, Two ways to use *#include* 4 8

3. LOOPS AND DECISIONS

Relational Operators; Loops:-The *for* Loop, *for* Loop Variations, The *while* Loop, Precedence : Arithmetic and Relational Operators, The *do* Loop, Uses.; Decisions:- The *if* Statement, The *if ... else* Statement, The *else... if* Construction, The *switch* Statement, The Conditional Operator.; Logical Operators:-The Logical *AND* Operator, The Logical *OR* Operator, The Logical *NOT* Operator. Precedence Summary.; Other Control Statements:-The *break* Statement, The *continue* Statement, The *goto* Statement 4 3

4. STRUCTURES

A Simple Structure, Specifying the Structure, Debugging a Structure Variable ,Accessing Structure Members, Other Structure Features, Structures within Structures, Structures and Classes, Enumerated Data Types 4 6

5. FUNCTIONS

Simple Functions:-The Function Declaration , Calling the Function, The Function Definition ,Comparison with Library Functions ,Eliminating the Declaration.

Passing Arguments to Functions:-Passing Constants, Passing Variables, Passing by Value, Passing Structure Variables, Names in the Declaration.

Returning Values from Functions:-The *return* Statement, Returning Structure Variables.; Reference Arguments.

overloaded Functions:-Different Numbers of Arguments, Different Kinds of Arguments.; Inline Functions; Default Arguments

h) Variables and Storage Classes:-Automatic Variables, External Variables, Static Variables, Storage.; Returning by Reference 4 5

6. OBJECTS AND CLASSES

Simple Class:- Classes and Objects, Specified the Class, Using the Class.

C++ Objects as Physical Objects; C++ Objects as Data Types; Constructors and Destructors

Object as Function Arguments:-Overloaded Constructors, Member Functions Defined Outside the Class, Objects as Arguments.

Returning Objects from Functions; Structures and Classes

Classes Objects and Memory; Static Class Data 4 6

7. ARRAYS

An Introduction to Arrays:- Defining Arrays, String Constants, Reading Embedded Blanks, Reading Multiple Lines, Arrays of Strings, Strings as Class Members, User Defined Strings. 3 5

8. OPERATOR OVERLOADING

Overloading Unary Operators

The Operator Keyword:-Operator Arguments, Operator Return Values, Nameless Temporary Objects, Limitations of Increment Operators.

Overloading Binary Operators:-Arithmetic Operators, Adding Polar Co-ordinates, Concatenating Strings, Multiple Overloading, Comparison Operators, Arithmetic Assignment Operators .

Data Conversion:-Conversions Between Basic Types, Conversion Between Objects and Basic Types, Conversion Between Objects of Different Classes, Conversions : When to Use What. Pitfalls of Operator Overloading and Conversion 3 5

9. INHERITANCE

Derived Class and Base Class:-Specifying the Derived Class, Accessing the Base Class Members, The protected Access Specifier.; Derived Class Constructors

Overriding member Functions; Class Hierarchies:-"Abstract" Base Class, Constructor and Member functions.

Public and Private Inheritance:-Access Combinations ,Classes and Structures, Usage of Access Specifiers. Levels Of Inheritance; Multiple Inheritance:-Member Functions in Multiple Inheritance, Constructors in Multiple Inheritance; Ambiguity in Multiple Inheritance; Inheritance and program Development 5 10

10. POINTERS

Addresses and Pointers:- The Address of Operator & Pointers Variables, Accessing the variable pointed to, Pointer to void.

Pointers and Arrays:-Pointers constraints and pointer Variables.

Pointers and Functions:-Passing Simple Variables, Passing Arrays, Sorting Array Elements.

Pointers and Strings:- Pointers to String Constants, Strings and functions Arguments, Copying a string using Pointers, Library String Functions, Arrays of Pointers to Strings.

Memory Management: new and delete:- The new Operator, The delete Operator, A String Class Using new.

Pointer to Objects:-Referring to members, Another Approach to new, An Array of Pointers to objects. Linked List.

Pointers to Pointers:-Sorting Pointers , Comparing Strings. Debugging Pointers. 4 8

11. FILES AND STREAMS

Steams:-The Steam Class Hierarchy, Stream Classes, Header Files.

String I/O:-Writing Strings, Reading Strings , Detecting End-of-File, Character I/O.

Object I/O:-Writing an Object to Disk, Binary Vs Character Files, Reading An Object from Disk, Compatible Data Structures.

I/O with Multiple Objects:-The *fstream* class, The *open* function.

File Pointers:-Specifying the position, Specifying the Offset, The *tellg* Function.

Disk I/O with Member Functions:-Closing Files, Other Data Structures.

Error Handling; Redirection:-*ios* Flags, Using *REDIR*, Redirecting Output, Redirecting Input,Redirecting Input and Output, The *cerr* and *clog* Objects.

Command- Line Arguments.; Printer Output.

Overloading the Extraction and Insertion Operators 6 10

12. C++ LIBRARY & MULTI- FILE PROGRAMS

String Class; Stack Class.; Containers Class Hierarchy:-Instance and Abstract Classes, Member Function; Data Class; Array Class:-Adding Elements to Arrays, Arrays Notation in Class Arrays, Inserting Array Elements; The List Class; The Queue Class:-The put and get Functions.; User - Define Classes; Discussion on Multi-File Programs in Class

13. GRAPHICS & VIRTUAL FUNCTIONS 6 15

Text- Mode Graphics Functions:-The window Function, Usage of Text -Mode Function, The cputs Function, The Clrscr Function; Graphics - mode Graphics Functions:-The initgraph Function, The circle Function, The closegraph Function

Colours:-setcolour Function, setlinestyle Function, setfillstyle Function

Rectangles and lines:-The ***rectangle()*** Function, The ***Line()*** Function

Virtual Functions, Friend Functions, Static Functions.

Total 54? 10

PRACTICALS:

The practical work in this course will be based on the series of programming exercises, which will reinforce the concepts of Object Oriented Paradigm (OOP). The programming will be implemented c Turbo C++ or Borland C++ Compiler. The List of Programming exercises will be based on the following constructs.

S.No	Item	No of turns	S.No	Item	No of tur
1	Input and output statement	1	2	Selection Structure	1
3	if statement	1	4	switch statement	1
5	Conditional operation statement	1	6	Repetition Structure	1
7	for statement	1	8	while statement	1
9	do - while statement	1	10	Functions	1
11	Structures	1	12	Objects and Classes	1
13	Arrays	1	14	Strings	1
15	Operator Overloading	1	16	Inheritance	1
17	Pointers	2	18	Polymorphism	1
19	Files and Streams	2	20	Graphics	2

REFERENCE BOOKS:

1. Object oriented Programming in C++, Robert Lafore, Galgotia.
2. The Anotated C++ reference Manual, Margaret Ellis & Bjarne Strousstrup.
3. An Introduction to the OOPS (Only Chapter 6), K.V.Witt,Galgotia.
4. Jamsa's 1001 C/C++ Tips, Kris Jamsa, Galgotia.2