SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total Credit
							End Sem	Class Work	Sessional Work	End Sem	
IT28001	OBJECT ORIENTED PROGRAMMING	3	0	2	3	1	70	30	40	60	4

PRE-REQUISITES: None

COURSE OBJECTIVES:

This course is intended to teach

1. The model of object oriented programming i.e. abstract data types, encapsulation, inheritance and polymorphism.
2. Fundamental features of an object oriented language like Java: object classes and interfaces, exceptions and various libraries.
3. To determine suitable logic for solving any given problem statement; then be able to proceed to code that logic as a program written in Java.

COURSE OUTCOMES: The student will be able to:

CO1 Explain object oriented language features and benefits and comparison with procedural language.

CO2 Explain java language constructs, keywords and write basic java programs.

CO3 Apply object oriented principles i.e. encapsulation, abstraction, inheritance and polymorphism and relationships.

CO4 Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes.

CO5 Use java standard API library to write complex programs.

CO6 Design event driven GUI and web related applications which mimic the real word scenarios.

THEORY:

COURSE CONTENTS:

UNIT 1: Review of object oriented concepts: comparison between procedural and object oriented language, features of object oriented language. Benefits of object oriented programming. Java language basics: variables, data types, operations, expressions, control structures and loops, methods, arrays and strings, static, final.

Unit 2: Basics of object orientation, Objects and Classes: defining classes for objects, identifying candidates for classes and objects, attributes and methods, achieving encapsulation, data hiding and abstraction, constructor and destructors, method overloading, visibility modifiers, Relationships: association, aggregation, composition, generalization and specialization.

UNIT 3: Object oriented analysis, Inheritance: hierarchical abstractions, base class object, subclass, member access rules, super uses, using final with inheritance, method overriding, abstract classes, multiple and multilevel inheritance. Concepts of interface, abstract class Vs interface, code reusability, Polymorphism: run time and compile time polymorphism. Modelling techniques, modelling polymorphism and inheritance by UML.

UNIT 4: Object Oriented Programming: components (packages), defining, creating and accessing a package, understanding CLASSPATH, importing packages, extensibility, robustness (exception handling), benefits of exception handling, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. Thread life cycle, creating threads, thread priorities, synchronizing threads, daemon threads, interthread communication, thread groups.

UNIT 5: I/O package: stream classes, byte stream and character stream, file operations, random access files, Applets: concepts of applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets. Object programming in windowed environments: AWT, working with collections and maps.

TEXT BOOKS RECOMMENDED:

1. Herbert Schildt, “Java 2: The Complete Reference”, 5th Edition, Tata McGraw-Hill.
2. Bruce Eckel, “Thinking in Java”, 4th Edition, Pearson Education.
3. Grady Booch, “Object Oriented Analysis and Design”, 2nd edition, Pearson Education

REFERENCE BOOKS:

1. James Rambaugh, Michael Blaha “Object Oriented Modeling and Design”, 2nd edition, Pearson Education.
2. Cay S. Horstmann and Gary Cornell, “Core Java: Volume I, Fundamentals”, 8th edition Prentice Hall.

ASSESSMENT TOOLS:
Direct assessment: End-Sem Examination, Mid-Term Test, Class Assignments, Quiz, Attendance
Indirect assessment: Course End Survey

PRACTICAL:
LAB ASSIGNMENTS:

Assignment 1

1. Write a program to display 'Hello World' as the output.
2. Write a program to calculate the sum of the digits of a five digit number
3. Write a program to reverse a number.
4. Write a program to display the following:

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

Assignment 2

1. Write a program to perform matrix multiplication.
2. For a given user input (integer value) compute the following sequence.

If number is even half it, if it is odd multiply by 3 and add 1.

Repeat the process until the value is 1. Print/Display all the values in the sequence.

Assignment 3

1. Write a program to find all prime numbers between two given numbers.
2. Write a program to find the minimum element in the given array and also search for a element in the given array and if present print its location.
3. Write a program to find the volume of cone by creating class with appropriate fields and methods.

Assignment 4

Create a class Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four piece of information as instance variables (a) part number (String type) (b) part description (String type) (c) quantity of the item being purchased (int type) (d) price per item (double type). The class should have a constructor that initialises the four instance variables. Provide set and get methods for each variables.

Assignment 5

Write a program to find the addition of two 'time data' objects of class. Time data is of the form hour, minutes and seconds. Take one data from user and another using constructor. For e.g. Given two time data 08:20:46 and 12:24:34, the resultant time will be 20:45:20.

Assignment 6

Consider an example of declaring the examination result. Design two classes 'Student' and 'Exam'. The student has data member's rollno, name and course and 'input_student' and 'display_student' as methods. Create the class Exam by inheriting Student class. The Exam class has fields Mark1, Mark2, Mark3 of three subjects and 'input_marks' and 'display_result' as member methods. Write an interactive program to model the relationship. Create an array of Exam objects and display result of n students, where n can be 1, 2, 3.....etc.

Assignment 7

Write a program (using exception handling) to get the data corresponding to age of a user and if the person is eligible to vote (determined with the help of method) display message correspondingly else if not eligible then to throw a user defined exception and handle it by displaying an appropriate message to the user.

Assignment 8

Write a program (using awt) to build a user login form consisting of controls enabling the user to input username and password details.

ASSESSMENT TOOLS:

Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test

Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO	PO	PO	PO	PO	PO	PO	PO	PO	PO	PO	PO	PSO	PSO	PSO
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	2	1	-	-	-	-	-	-	-	-	-	1	1	-
CO2	3	3	1	2	-	-	-	-	-	-	-	-	3	2	-
CO3	3	3	3	2	-	-	-	-	-	-	-	-	2	2	-
CO4	3	3	3	-	-	-	-	-	-	-	-	-	2	3	-
CO5	3	2	2	-	-	-	-	-	-	-	-	-	2	3	1
CO6	3	2	2	-	-	-	-	-	-	-	-	-	2	3	-

SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total Credit
							End Sem	Class Work	Sessional Work	End Sem	
IT28007	DIGITAL SYSTEM	3	0	2	3	1	70	30	40	60	4

PRE-REQUISITES: Fundamentals knowledge of digital circuits.

COURSE OBJECTIVES:

This course is intended to

1: Introduce the concept of digital circuits and number systems.

2: Provide students to be able to design and analyze combinational and sequential logic circuits.

COURSE OUTCOMES:

After completion of the course, students will be able to:

CO1 Explain architecture of basic building blocks of digital circuits.

CO2 Solve the problems of number system and logic gates.

CO3 Design logic circuits using Boolean algebra.

CO4 Design digital components like decoders, multiplexers, arithmetic circuits and other combinational circuits.

CO5 Analyse design of combinational and sequential circuits.

CO6 Describe the various types of Programmable Logic Devices.

THEORY:

COURSE CONTENTS:

Unit1: Introduction and type of Number system- Binary, Octal, Decimal and Hexadecimal Number System, Conversion between number system, Arithmetic operations on number system, Signed and unsigned number system. Digital signal, logic gates: AND, OR, NOT, NAND, NOR, EX-OR, EX-NOR, Boolean algebra, Binary codes: BCD, Excess-3, Gray, EBCDIC, ASCII, Error detection and correction codes**.**

Unit2: Boolean postulates and laws, De-Morgans Theorem, Principle of Duality, Boolean expression, Minterm, Maxterm, Sum of Products (SOP), Product of Sums (POS), and K- map Minimization: Don’t care conditions, Quine-McCluskey method of minimization. Variable Entered Maps, Realizing Logic Function with Gates.

Unit3: Design procedure Half adder, Full Adder, half subtractor, Full subtractor, Parallel binary adder &amp; Subtractor, Fast Adder, Carry Look Ahead adder, Serial Adder/Subtractor, BCD adder; Binary Multiplier, Binary Divider, Multiplexer/ Demultiplexer, decoder, encoder, parity checker, parity generators - code converters - Magnitude Comparator.

Unit4: Latches, Flip-flops - Characteristic table and equation, Application table, Edge triggering, Level Triggering, Realization of one flip flop using other flip flops; Asynchronous counters, Synchronous counters, Programmable counters, Design of Synchronous counters: statediagram State table, State minimization, State assignment, Excitation table and maps; Circuit implementation, Modulo n counter, Registers shift registers - Universal shift registers, Shift register counters, Ring counter, Shift counters, Sequence generators.

Unit5: Introduction to Programmable Logic Devices, Read-Only Memory, Programmable Logic Arrays (PLA), Programmable Array Logic (PAL), Combinational PLD-Based State Machines, State Machines on a Chip.

TEXT BOOKS RECOMMENDED:

1. Digital Design, M. Morris Man, Prentice Hall of India Pvt. Ltd.
2. Digital Design, John F. Wakerly, Pearson/PHI.

REFERENCE BOOKS:

1. Digital Circuits and Design, S. Salivahanan and S. Arivazhagan; VikasPublishing House Pvt. Ltd, New Delhi.
2. DIGITAL DESIGN, JOHN F. WAKERLY, PEARSON/PHI.

ASSESSMENT TOOLS:

Direct assessment: End-sem examination, mid-semester test, class assignments, attendance

Indirect assessment: Course End Survey

PRACTICALS

Assignment 01
Programming concept for convert the binary number to octal number system.
Programming concept for convert binary number to hexadecimal number system.
Programming concept for convert decimal number to hexadecimal number system.

Assignment 02
Programming concept for convert the decimal number to binary number system.
Programming concept for convert the binary number to decimal number system.
Programming concept for convert the decimal number system to octal number.
Programming concept for convert octal number system to decimal.

Assignment 03

Compute and implement the addition and subtraction using complements for the following bases.

1. Base 2
2. Base 8
3. Base 10
4. Base 16

Assignment 04
Implement the functions that can predict the given number is in the range or not. If it is given in the range then show the output as number is in the range otherwise output would be number is not in the range. In this we take the number which can be test and take the number of digit i.e. n in the program.
There are three cases of the number system:

1. Signed magnitude The range of signed magnitude is -(2n1-1) to (2n-1-1)
2. 1’s compliment The range of 1’s compliment is -(2n-1- 1) to (2n-1-1)

3. 2’s compliment The range of 2’s compliment is -(2n-1) to (2n-1-1)

Assignment 05
Program to find 1’s & 2’s complement of a binary number.
Program to find 7’s & 8’s complement of an octal number.
Program to find 9’s & 10’s complement of a decimal number.
Program to find 15’s & 16’s complement of a hexadecimal number.

Assignment 06
Assignment is based on the Boolean algebra operations implement. You can take two number as an input. Save these inputs to the variables. Convert these into the Boolean number. Perform the following logical operation on these two numbers.

1. AND operation
2. OR operation
3. NOT operation
4. NAND operation
5. NOR operation
6. XOR operation
7. XNOR operation

ASSESSMENT TOOLS:
Direct assessment: Lab assignments, Quiz, project presentation, Viva-voce examination, attendance.
Indirect assessment: Course End Survey, External Examinar Feedback.

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or ‘-’):

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	2	-	1	-	-	-	-	-	-	-	-	1	-	1	-
CO2	3	2	1	1	-	-	-	-	-	-	-	2	1	-	1
CO3	3	2	2	1	-	-	-	-	-	-	-	2	2	2	1
CO4	3	3	1	1	-	-	-	-	-	-	-	2	2	3	-
CO5	1	2	1	2	-	-	-	-	-	-	-	2	1	2	2
CO6	1	1	1	1	-	-	-	-	-	-	-	2	-	-	1
AVG	2.16	2	1.16	1.2	-	-	-	-	-	-	-	1.83	1.5	2	1.25

SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total Credit
							End Sem	Class Work	Sessional Work	End Sem	
IT28008	DATA STRUCTURES	3	0	2	3	1	70	30	40	60	4

PRE-REQUISITES: Fundamentals knowledge of programming and C language.

COURSE OBJECTIVES:

1. To provide the knowledge of linear and non-linear data structures.
2. To develop skills to apply appropriate data structure in problem solving.

COURSE OUTCOMES:

After completion of the course, students will be able to:

CO 1 **Describe** various linear and non-linear data structures.
CO 2 **Analyze** running time complexity of algorithms using asymptotic analysis (big-O notation)
CO 3 **Solve** problems involving graph and tree.
CO 4 **Apply** sorting and searching algorithms to the small and large data sets
CO 5 **Describe** the hash function and concepts of collision and its resolution methods
CO 6 **Choose** appropriate data structures to solve real world problems efficiently.

THEORY:

COURSE CONTENTS:

UNIT 1: Introduction to Data Structures: Data types, abstract data types, Arrays: definition, single dimensional and multidimensional arrays, strings, Linked List: definition and application, singly linked lists, doubly linked lists, and circular list, sparse matrices, application of linked lists. Algorithms efficiency and big-O notation.

UNIT 2: Stacks: definition, stack implementation and operations, infix, postfix and prefix expressions, stack application, recursion, recursive definition and processes, recursive examples(Tower of Hanoi, Fibonacci Series etc),Recursion Types(Head Recursion ,Tail Recursion, Excessive Recursion, Direct Recursion, Indirect Recursion), Queues: definition, queues implementation, sequential representation, circular queue, dequeue, and queue application.

UNIT 3: Tree: binary tree, Binary Tree representations: node and implicit array representation, internal and external nodes, binary tree traversals, threaded binary trees, Representing list as binary trees, Huffman algorithm, heterogeneous binary trees, Tree Searching: insertion into and deletion from binary search tree, efficiency of binary search tree operations, AVL tree, red-black tree and tree applications.

UNIT 4: Graph: Graphs, representation of graphs using adjacency matrix and adjacency linked lists, Spanning Trees, Graph Traversals: depth first search and breadth first search, Dikstra's Shortest path algorithm. General search trees : multiway search trees , searching, implementing and traversing a multiway search tree, B-tree, binominal heap, Fibonacci heap, B+ Tree, Digital search trees.

UNIT 5: Searching & Sorting: Dictionary as ADT, efficiency of sequential searching, searching an ordered table, indexed sequential search, Binary search: general background, efficiency consideration, Exchange sorts: bubble sort, quick sort, selection and tree sorts, straight selection sort, binary tree sort, heap sort, insertion sorts: simple insertion, shell sort, merge and radix sorts. Hashing: open hashing; close hashing, rehashing, dynamic and extendible hashing.

TEXT BOOKS RECOMMENDED:

1. Langsum, Augestein and Tenenbaum,"Data structures using C", 2nd Edition Pearson Education.
2. Thomas H. Cormen ,"Introduction to algorithm", 2nd Edition, Prentice Hall of India.
3. Mark Allen Weiss,"Data structures and algorithm analysis in C", 3rd Edition, Pearson Education.

REFERENCE BOOKS:

1. Yeshwant Kanetkar , "Data Structures through C", 1st Edition BPB.
2. Behroz A. Fourozan, "Data Structure: A Pseudocode approach", 2nd Edition, Cengage Learning India.

ASSESSMENT TOOLS :

Direct assessment: End-Sem Examination, Mid-Term Test, Class Assignments, Quiz, Attendance
Indirect assessment: Course End Survey

PRACTICAL:

LAB ASSIGNMENTS:

1. Write a program to
   a) Create dynamic int array using malloc() and free()

   (b) Create dynamic char array using calloc() and free().

2 Write a program to implement 1) linear Search 2) Binary Search.

3 Write a program to implement 1) Bubble Sort 2) Insertion Sort 3) Selection Sort 4) merge sort and 5) Quick sort.

4 Write a menu driven program to implement following operations on the singly linked list. (a) Insert a node at the front of the linked list. (b) Insert a node at the end of the linked list. (c) Insert a node such that linked list is in ascending order.(according to info. Field) (d) Delete a first node of the linked list. (e) Delete a node before specified position. (f) Delete a node after specified position.

5 Write a program to implement following operations on the doubly linked list. (a) Insert a node at the front of the linked list. (b) Insert a node at the end of the linked list. (c) Delete a last node of the linked list. (d) Delete a node before specified position.

6 Write a program to implement following operations on the circular linked list. (a) Insert a node at the end of the linked list. (b) Insert a node before specified position. (c) Delete a first node of the linked list. (d) Delete a node after specified position.

7 Write a program for stack that performs following operations using array. (a) PUSH (b) POP (c) PEEP (d) CHANGE (e) DISPLAY.

8 Write a program to convert infix notation to postfix notation using stack.

9 Write a program to implement QUEUE using link list that performs following operations (a) INSERT (b) DELETE (c) DISPLAY.

10 Write a program to implement circular Queue using array that performs following operations (a) INSERT (b) DELETE (c) DISPLAY.

11 Write a program to implement stack using link list.

12 Write a program to implement Queue using link list.

13 Write program to create binary Tree and create binary Tree Traversal.

ASSESSMENT TOOLS:
Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test
Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	3	2	1	-	-	-	-	-	-	-	-	1	3	2	2
CO2	3	3	1	2	2	-	-	-	-	-	-	1	-	1	1
CO3	3	3	2	2	-	-	-	-	-	-	-	1	1	2	1
CO4	3	2	2	-	-	-	-	-	-	-	-	1	3	1	1
CO5	3	2	2	-	-	-	-	-	-	-	-	1	2	1	1
CO6	3	3	3	3	1	1	-	-	-	-	-	1	3	3	1

SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
DEPARTMENT OF INFORMATION TECHNOLOGY

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total Credit
							End Sem	Class Work	Sessional Work	End Sem	
IT28498	System Software Laboratory-I	-	-	2	-	1	-	-	40	60	1

PRE-REQUISITES: None

COURSE OBJECTIVES:

1. Describe the basic file system in Linux and its file attributes.
2. Summarize different Linux commands to write Shell Programs.
3. Use different filters, process handling, regular expressions features using suitable commands.
4. To understand and make effective use of linux utilities and shell scripting language to solve problems

COURSE OUTCOMES: After completion of course, students will be able to:

CO1 **Demonstrate** installation of Linux operating system.
CO2 **Work** within a command line environment.
CO3 **Create** files and directories and operate them.
CO4 **Write** shell scripts using vi editor.
CO5 **Apply** the concept of filter using regular expression.
CO6 **Manage** shell and processes using various commands.

THEORY:
COURSE CONTENTS:

UNIT 1: Introducing to UNIX / Linux: The Operating System , UNIX Operating System, knowing Your Machine, A Hands-On Session: System Information with date and who, Viewing Processes with ps, Handling Files, common UNIX / Linux Commands, Handling Directories, UNIX Architecture.

UNIT-II: The File System: File concepts, File types File system structure, file metadata - Inodes, File System hierarchy, creating, removing and changing Directories-mkdir, rmdir, chdir, obtaining current working directory-getcwd, directory contents, scanning directories- opendir, readdir, rewin functions.

UNIT-III: The Editors: vi Basics, Input Mode—Entering and Replacing Text, Saving Text and Quitting, Navigation, Editing Text with Operators, Copying and Moving Text from One File to Another, Undoing Last Editing Instructions, Searching for a Pattern ,Repeating the Last Pattern Search, Repeating the Last Command.

UNIT-IV: The Shells: Bourne Shell, C Shell, Shell Variables, Scripts, Meta Characters and Environment, if and case Statements, for, while and until loops. Awk Pattern Scanning and Processing, begin and end Patterns, Awk Arithmetic and Variables, builtin functions and Operators, Arrays, Strings.

UNIT-V: Filters Using Regular Expressions: Simple Filters: Pr, cmp, comm, diff, head, tail, cut, paste, sort, uniq, tr, The Sample Database: grep: Searching for a Pattern, Quoting in grep, When grep Fails, grep Options, Basic Regular Expressions (BRE)—An Introduction, Character Class.

TEXT BOOKS RECOMMENDED:

1. "UNIX-Concepts and Applications", Sumitabha Das 4th Edition, Tata McGraw Hill, 2006.
2. Unix System Programming using C++, T.Chan, PHI

REFERENCE BOOKS:

1. Begining Linux Programming, 4th Edition, N. Matthew, R.Stones, Wrox, Wiley India Edition.
2. Unix for Programmers 3rd Ed, Graham Glass & King Ables, Pearson Education

PRACTICAL:

LAB ASSIGNMENTS:

1. Introduction- Linux Architecture- Shell, Kernel, System calls.

Linux installation- Steps for installing Linux Operating System Comparison between Linux and other Operating Systems, Applications of Linux Operating System.

2. Internal & External commands in Linux.
   - Internal commands- echo, type, etc.
   - External commands- ls, cp, mv, rm, cat, etc
   - Other commands – tput clear, who, cal, date, bc, man, passwd, uname.
3. Working with files & directories.
   - Know the categories of files.
   - Directory related Commands – pwd, mkdir, rmdir, cd, ls
   - Manipulating Absolute paths and Relative paths using cd command.
   - File related Commands – cat, cp, mv, rm, comm, cmp, diff, tar, umask, wc
4. Basic File attributes.
   - Listing seven attributes of a file : ls and its options
   - File Permissions: Absolute and Relative permissions
   - Manipulating File permissions using chmod command
   - Manipulating File Ownership using chown command
5. Learn to use vi editor.
   - Three modes of vi editor.
   - Input mode commands.
   - Command mode commands.
   - Ex mode commands.

6. Simple Filters – head, tail, cut, paste, sort, uniq, tr, pr.
7. Expressions & search patterns .(dot operator), *, ^, +, ?, grep, egrep, fgrep
8. Introduction to shell programming.
   - Introduction, Uses of shell script, Shell special characters, comments, command separator, escaping, quoting command substitution.
   - Creating shell script, Shell identifiers, Shell variables, Destroying a variable, Positional parameters & command line arguments.
9. Shell control structures
   - if, case, for, while, relational and logical operators.

ASSESSMENT TOOLS:
Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test
Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	1	1	1	-	3	1	-	-	-	-	-	1	-	-	-
CO2	1	2	3	2	-	-	-	-	-	-	-	-	-	2	-
CO3	1	1	-	-	-	-	-	-	-	-	-	-	-	1	-
CO4	3	2	3	1	-	-	-	-	1	-	-	-	2	1	1
CO5	2	1	2	1	-	-	-	-	-	-	-	-	-	1	1
CO6	1	1	1	1	-	-	-	-	-	-	-	-	-	-	1