Academic Council 25/05/2011
Item No. 4.90

UNIVERSITY OF MUMBAI

Syllabus for the F.Y.B.Sc.
Program: B.Sc.
Course : Computer Science

(Credit Based Semester and Grading System with effect from the academic year 2011–2012)

F.Y. B.Sc. Syllabus
(Credit, Grade and Semester System)
To be introduced from the Academic Year 2011 – 2012

Computer Science – Single Major Course

The credits earned by the learner in the duration of the three year undergraduate programme in Computer Science is shown in the following Table, assuming that the student has taken Computer Science, Physics, Mathematics and Foundation courses in the first year, Computer Science, Mathematics, and Foundation courses in the second year and Computer Science and Applied Component in the third year.

For Course per week 1 lecture/period is 48 minutes duration				**For subject per week** 1 lecture/period is 48 minutes duration			
	Theory	Practical	Tutorial		Theory	Practical	Tutorial
Actual Contact	3	3	-	Actual Contact	6	6	-
Credits	2	1	-	Credits	4	2	-

Year	**Sem**	**Computer Science**		**Mathematics**		**Physics**		**FC**	**AC**		**Total**
		Th	Pr	Th	Pr	Th	Pr	Th	Th	Pr	
	I	4	2	4	2	4	2	2			20
	II	4	2	4	2	4	2	2			20
	III	6	3	6	3			2			20
	IV	6	3	6	3			2			20
	V	10	6						2	2	20
	VI	10	6						2	2	20
Total		46		30		12		8	8		120

Course Code	Title	Credits
USCS101	**COMPUTER ORGANIZATION -1**	**2 Credits** **(45 lectures)**
Unit I : Introduction: **(a) Computers:** History of computers and their classification **(b) Basics of modern computer systems:** View of a computer as an integrated system, Neumann machine, block diagram of a computer system. **(c) Information:** Definition, Characteristics and interpretation, Data and its logical and physical concepts, binary form of program and instruction. **(d) Number Systems:** Binary, Decimal, Octal, Hexadecimal and their inter-conversions. **(e) Computer Arithmetic:** Binary addition and subtraction using signed-Magnitude, 1's complement and 2's complement, Binary multiplication and division, Floating point representation and arithmetic, arithmetic through stacks. **(f) Codes for character representation:** hexadecimal, BCD, Excess-3, Gray code, ASCII, EBCDIC, Unicode.		**15 Lectures**
Unit II : Digital logic circuits: **(a) Boolean algebra:** Basic identities of Boolean Algebra, Boolean function **(b) Logic Gates:** AND, OR, NOT, NOR, NAND, EX-OR EX-NOR operations and their truth table, Minimization of gates by K-maps. **(c) Digital Circuits:** Half Adder, Full Adder, Binary adder-subtractor, binary incrementer, Multiplexers, Encoder and decoder. **(d) Flip Flops:** Concept of sequential circuits, concept of clock and synchronization, S-R, J-K, Preset and Clear, Master-Slave J-K, D, T Flip Flops, their truth tables and identities, Conversion from one type to another type of Flip Flop, concept of counters and registers, shift registers		**15 Lectures**
Unit III : Introduction to computer components: **(a) Memory:** Primary Memory – RAM, SRAM, DRAM, ROM, EPROM. Secondary Memory – Magnetic Floppy and Hard Disk. Optical Memory – CDROM, WORM, Concept of Virtual Memory, Concept of Cache and their		**15 Lectures**

need, Memory hierarchy. **(b) Input/output devices**: Input/output devices, input/output interface, asynchronous data transfer, modes of data transfer.. **(c)CPU:** Functions of CPU, register classification and organization, instruction sets and examples of instruction set, addressing schemes, instruction formats, instruction cycle and instruction pipelining.	

Course Code	Title	Credits
USCS102	**ALGORITHMS AND PROGRAMMING IN C – 1**	**2 Credits** **(45 lectures)**
Unit I : Introduction to algorithms and C programming: **(a) Fundamentals of algorithms:** Notion of an algorithm. Pseudo-code conventions like assignment statements and basic control structures. **(b) Algorithmic problems:** Develop fundamental algorithms for **(i)** Exchange the values of two variables with and without temporary variable, **(ii)** Counting positive numbers from a set of integers, **(iii)** Summation of set of numbers, **(iv)** Reversing the digits of an integer, **(v)** Find smallest positive divisor of an integer other then 1, **(vi)** Find G.C.D. and L.C.M. of two as well as three positive integers, **(vii)** Generating prime numbers. **(c) Analysis of algorithms:** Running time of an algorithm, worst and average case analysis. **(d) Different approaches in programming:** Procedural approach, Object Oriented approach, Event Driven approach. **(e) Structure of C**: Header and body, Use of comments, Compilation of a program. **(f) Data Concepts:** Variables, Constants, data types like: int, float char, double and void. Qualifiers: short and long size qualifiers, signed and unsigned qualifiers. Declaring variables, Scope of the variables according to block, Hierarchy of data types.		**15** **Lectures**
Unit II : Basics of C programming: **(a) Types of operators:** Arithmetic, Relational, Logical, Compound Assignment, Increment and decrement, Conditional or ternary, Bitwise and Comma operators. Precedence and order of evaluation., Statements and Expressions. **(b) Type conversions:** Automatic and Explicit type conversion. **(c) Data Input and Output functions:** Formatted I/O: printf(), scanf(). Character I/O format: getch(), getche(), getchar(), getc(), gets(), putchar(), putc(), puts(). **(d) Iterations:** Control statements for decision making: **(i)** Branching: if statement, else.. if statement, switch statement. **(ii)** Looping: while loop, do.. while, for loop. **(iii)** Jump statements: break, continue and goto		**15** **Lectures**
Unit III : Arrays, Strings, Structures and Storage Classes: **(a) Arrays:** (One and multidimensional), declaring array variables, initialization of arrays, accessing array elements. **(b) Strings:** Declaring and initializing String variables, Character and string handling functions. **(c)Structure:** Declaration of structure, reading and assignment of structure variables, Array of structures, arrays within structures, structures within structures **(d) Unions**: Defining and working with union. **(e)Storage classes:** Automatic variables, External variables, Static variables, Register variables.		**15** **Lectures**

Course Code	Title	Credits
USCSP1	**PRACTICALS**	**2 Credits**

SECTION – I

(A) Introduction to Operating system desktop, folders, files, shortcuts, popular menus, using notepad, word, excel, power point.

(B) Introduction to windows wildcard characters, absolute path, relative path and commands like md, cd, rd, copy, ren, del etc.

(1) Demo practical on various internal and external parts of computer and their interconnection/working.

(2) Demo hands on assembly of PC.

(3) Study of basic gates.

(4) Implementation of Boolean equations using basic gates.

(5) Study of flip-flops.

(6) Study of 4 to 1 multiplexer.

(7) Study of decoder

(8) Study of counters.

(9) Study of universal shift registers.

(10) Study of 4 bit adder/ Subtractor.

Note: (1) (1) Practical A and B are compulsory. They are to be written in jjournal but should not be the part of practical examination.

(2) In all **Eight** practical (including A and B) from the list should be performed.

45 Lectures

SECTION - II

Suggestions while writing programs in C:

(i) Use of comments at appropriate places is necessary.

(ii) Use appropriate indentation while nesting the loops, if-else statements.

(iii) Do not ignore warnings after the compilation.

(iv) Optimize the codes as far as possible, by using optimization techniques.

Following is the list of suggested practical in C:

(1) Convert the following algorithms using C: Exchange the values of two variables with and without temporary variable.

(2) Convert the following algorithms using C: Counting positive numbers from a set of integers.

(3) Convert the following algorithms using C: Summation of set of numbers.

(4) Convert the following algorithms using C : Reversing the digits of an integer.

(5) Convert following algorithms using C: Find smallest positive divisor of an integer other than 1.

(6) Convert the following algorithms using C: Find G.C.D. and L.C.M. of two as well as three positive integers.

(7) Convert the following algorithms using C: Generating prime numbers.

(8) Write a program to find the **(a)** sum of two matrices of order m × n and transpose of order m × n where m, n ≤ 3. **(b)** multiplication of two matrices of order m, where m ≤ 3, finding square and cube of a square matrix. **(c)** Inverse of a matrix**(d)** |A|, |B| and verify the identity |A B| = |B A | = |A| |B| , where | | denote determinant of the matrix and A and B of size 2 × 2.

(9) Write a program to **(a)** input a sentence **(b)** count the number of occurrences of the given pattern of letters (for instance ‘est or ‘ed’) **(c)** find the position of the rightmost or leftmost character occurred from the pattern of letters

(10) Write a program which counts the number of **(a)** paragraphs occurred.**(b)** times the word “the” appears in a short story

Write a program to create structure to **(a)** find and print the average marks of five subjects along with the name of student. **(b)** store names of the states (within India) and their capital cities. Show the capital by inserting state from the keyboard.

Course Code	Title	Credits
USCS201	**COMPUTER ORGANIZATION -2**	**2 Credits** **(45 lectures)**
Unit I : Memory and Device organization: **(a) Internal memory organization:** DRAM, SRAM, ROM types, Cache Memory Principles, elements of cache design, Pentium 4 cache. **(b) External memory organization:** Magnetic disk, RAID, Optical memory, Magnetic tape **(c) Input/Output device organization:** External devices, I/O modules, Concepts of programmed I/O, interrupt Drive I/O, DMA, I/O processors.		**15 Lectures**
Unit II : Operating System Support and Introduction to multiprocessors: **(a) Operating System Support**: Basic Concepts, Batch, Multiprogramming and Time-Sharing, scheduling, scheduling, Memory Management. **(b) Introduction to multiprocessors:** Characteristics of Multiprocessors, Time-Shared Bus, Multi-port memory.		**15 Lectures**
Unit III : Introduction and programming with Microprocessors: Introduction to 8085 Architecture and its extension to architecture to 8086, functional block diagram, Bus interface unit, Execution unit, general purpose registers, segment registers, pointers and index registers basic instruction set and organization of 8086, 8086 Assembly language programming		**15 Lectures**

Course Code	Title	Credits
USCS202	**ALGORITHMS AND PROGRAMMING IN C – 2**	**2 Credits** **(45 lectures)**
Unit I: Functions, Recursion and Sorting: **(a) Functions:** Global and local variables, Function definition, return statement, Calling a function by value, Macros in C, Difference between functions and macros. **(b) Recursion:** Definition, Recursion functions algorithms for factorial, Fibonacci sequence, Tower of Hanoi. Implement using C. **(c) Sorting Algorithms:** Bubble, Selection, Insertion and Merge sort, Efficiency of algorithms, Implement using C.		**15** **Lectures**
Unit II: Pointers and File handling **(a) Pointer:** Fundamentals, Pointer variables, Referencing and de-referencing, Pointer Arithmetic, Chain of pointers, Pointers and Arrays, Pointers and Strings, Array of Pointers, Pointers as function arguments, Functions returning pointers, Pointer to function, Pointer to structure, Pointers within structure. **(b) Dynamic Memory Allocation**: malloc(), calloc(), realloc(), free() and sizeof operator. **(c) File handling:** Different types of files like text and binary, Different types of functioms:fopen(), fclose(), fgetc(), fputc(), fgets(), fputs(), fscanf(), fprintf(), getw(), putw(), fread(), fwrite(), fseek()		**15** **Lectures**
Unit III: Stacks, Linked Lists and Queues **(a) Stacks:** Definition, Array representation of stacks, Algorithms for basic operators to add and delete an element from the stack, Implement using C. **(b) Linear Link lists:** Representation of link list in memory, Algorithms for traversing a link list, searching a particular node in a link list, insertion into link list (insertion at the beginning of a node, insertion after a given node), deletion from a link list. Implement using C. **(c) Queues:** Representation of queue, Algorithm for insertion and deletion of an element in a queue, Implement using C.		**15** **Lectures**

<table>
<tr><th>Course Code</th><th>Title</th><th>Credits</th></tr>
<tr><td>**USCSP2**</td><td>**PRACTICALS**</td><td>**2 Credits**</td></tr>
<tr><td colspan="2">**COMPUTER ORGANIZATION -2**
(1) Demo practical on working of 8085.
(2) Study of internal memory, I/O modules.
(3) Study of operating system.
(4) Study of networking of computers and other devices.
(5) Study of concepts of parallel processing.
(6) Study of 8086 architecture.
(7) Study of 8086 instruction set
(8) Writing programs with 8086 microprocessor for
(a) Addition of 1 to n numbers
(b) Finding largest/smallest from n given numbers.
(9) Writing program with 8086 microprocessor for demonstration of use of JUMP instructions.
(10) Writing programs with 8086 microprocessor for
(a) Use of I/O ports.
(b) Block transfer of memory.
Note: (1) **<u>Any five</u>** experiments from 1 to 7 practical should be performed.
(2) Experiments from 8 to10 are compulsory.</td><td>**45 Lectures**</td></tr>
<tr><td colspan="2">**ALGORITHMS AND PROGRAMMING IN C – 2**
(1) Write a program to create functions **(a)** to generate twin primes in a given range of numbers, **(b)** to find the prime factors of a given integer.
(2) Write a program to accept details of 5 customers that includes customer number, name and mobile number. Create a menu with options ‘Modify’, ‘Display’ and ‘Exit’. Write functions modify(), which will allow modification of mobile number and function display(), which will display all the details of customers.
(3) Given an array S of n integers. Write a program to **(a)** sort the elements in S in ascending order by considering an array of n-2 elements by any sorting method, **(b)** find the median of elements of S
(4) Write a program using pointer notation **(a)** to write function to exchange two strings, **(b)** to determine whether the given string is a palindrome, **(c)** to find the average of each students in 3 tests. Number of students can be given from keyboard.
(5) Write a program that accepts a number from the user and passes a pointer to the number to a function for processing. This function passes a pointer to this pointer to another function for processing of the number. Both the functions should display the number.
(6) Write a function called increment that accepts a date structure with three fields. The first field contains the month (a pointer to string). The second field is an integer showing the day in month. The third field is an integer showing the year. The function should increments the date by 1 day and returns the new date. If the date is the last date in the month, the month field must be changed. If the month is December, the value of year must be changed when day is 31. A year is leap year if **(a)** It is evenly divisible by 4 but not with 100 **(b)** It is evenly divisible by 400
(7) Write a program to crate a dynamic one and two dimension array by accepting number of rows and/or columns from the user at runtime using pointer notation.
(8) Write a program to **(a)** read string from the user to check whether it exists in a given file or not. **(b)**to accept a file name and then accept contents that should be</td><td>**45 Lectures**</td></tr>
</table>

stored in the file until the user types “end”, **(c)** declare a structure representing student, accept data of 5 students and store it in a file, **(d)** convert a capital case letter file to small case letter file.

(9) Write a program to compute factorial of a large number.

(10) The Ackerman’s function Ack(m ,n) is defined recursively by **(a)** If $m = 0$ then Ack(m ,n) = n+1, **(b)** If $m \neq 0$ but $n = 0$ then Ack(m ,n) = Ack(m-1 ,1), **(c)** If $m \neq 0$ and $n \neq 0$ then Ack(m ,n) = Ack(m-1 , Ack(m, n-1)) **(d)** Write a program to calculate Ack(0 ,n), Ack(4 ,0), Ack(6 ,10)

(11) Fibonacci sequence is defined by : **(a)** If $n = 0$ or $n = 1$ then $F_n = n$; **(b)** If $n > 1$ then $F_n = F_{n-2} + F_{n-1}$ **(c)** Write a program to generate Fibonacci sequence.

(12) Write a program to solve Tower of Hanoi problem

(13) Write a program to create a linked list containing student’s name and marks **(a)** search marks of a particular student **(b)** insert a new student at the beginning of the list. **(c)** insert a new student at the end of the list **(d)** delete a particular student from a list.

(14) Write a program to create following stack of characters, where STACK is allocated N=8 memory cells:

STACK: A, C, D, F, K, ___, ___, ___.

(For notation convenience, we use “__“ to denote an empty memory cell).

The program should describe the STACK after the following operations take place:

(a) POP (STACK, ITEM)
(b) POP(STACK, ITEM)
(c) PUSH(STACK, L)
(d) PUSH(STACK, P)
(e) POP (STACK, ITEM)
(f) PUSH(STACK, R)
(g) PUSH(STACK, S)
(h) POP(STACK, ITEM)

The POP always deletes the top element from the stack, and the PUSH always adds the new element to the top of the stack.

(15) Write a program to create following queue of cities, where QUEUE is allocated 6 memory cells:

QUEUE: _____, Latur, Bashirabad, Rawalgav,Palanpur, _____

(For notation convenience, we use “___ “to denote an empty memory cell).

Where FRONT = 2, REAR= 5.The program should describe the QUEUE, including FRONT and REAR, after the following operations take place:

(a) Allahabad is added, **(b)** two cities are deleted, **(c)** Mumbai is added, **(d)** Mathura is added, **(e)** three cities are deleted and **(f)** Nagpur is added.

REFERENCES

COMPUTER ORGANIZATION:

(1) Computer organization and architecture: William Stallings, PHI, Sixth edition

(2) Computer System architecture: M. Morris Mano, PHI,

(3) Microprocessor architecture, programming and applications with 8085: Ramesh Gaonkar, Fourth edition, Penram international.

(4) 8086 Microprocessor: Kenneth Ayala

ALGORITHMS AND PROGRAMMING IN C :

(1) Introduction to Algorithms (Second Edition): *Cormen, Leiserson, Rivest, Stein PHI(Chapter 1,2,3,10)*

(2) Data Structures (Schaum's outline series in computers): Seymour Lipschutz McGraw-Hill book Company (Chapter 2, 5, 6, 9)

(3) Programming in ANSI C (Third Edition) *: E Balagurusamy,* TMH *(Chapters: 2, 3, ,4, 5, 6, 7, 8, 9, 10, 11, 12, 13)*

Additional References:

(1) Fundamental Algorithms (Art of Computer Programming Vol 1): Knuth , Narosa Publishing House

(2) Mastering Algorithms with C, Kyle Loudon, Shroff Publishers

(3) Algorithms in C (Third Edition): Robert Sedgewick , Pearson Education Asia

(4) Data Structures A Pseudocode Approach with C: Richard F. Gilberg, Behrouz A. Forouzan , Thomson

(5) Let us C by Yashwant Kanetkar, BPB

(6) Programming in ANSI C by Ram Kumar, Rakesh Agrawal, TMH

(7) Programming with C (Second Edition): Byron S Gottfried (Adapted by Jitender Kumar Chhabra) Schaum's Outlines (TMH)

(8) Programming with C : K R Venugopal, Sudeep R Prasad TMH Outline Series.

(9) Unix and C : M.P. Bhave and S.A. Pateker, Nandu printers and publishers private limited.

Allocation of time per credit: 1 Credit = 30 to 40 hours

Total contact hours: 468 hours per Semester i.e. 936 hours per year

Ratio of instruction: Self study :– (i) Theory – 1:1, (ii) Practical – 4:1

The time duration per credit is divided into two parts:

1. Approximately fifty percent of the time will be spent on classroom instruction including practical as prescribed by the University.
2. Rest of the time spent as notional hours (30-40 hrs/credit)

(Notional Hours: Module to be selected as per the Department requirements.)

- Training for Assignment writing ,extra coaching for vernacular students, Journal writing
- Student seminars or group discussion
- Organize lectures or talks on the related subject.
- Organize open day in the department with the participation of FY students for junior college students
- Discuss career opportunities
- Counselling Lecture
- Industrial Visit, relevant to the subject
- CD Shows/Film shows
- E- book learning
- Visit to an NGO, Science exhibition
- Training for participation in extra -curricular activities.
- Interaction with parents.
- Attending seminars, workshop,& conferences
- Group activity/Self Study/Quiz.

Credit Assignment:

Semester I:

Course	Learning Hours(h) Lectures (L)		Credits	
	Theory	Practical	Theory	Practical
I (USCS 101)	45 L = 36 h	-	2	-
I (USCSP 101)	-	45 L = 36 h	-	1
II (USCS 102)	45 L = 36 h	-	2	-
II (USCSP 102)	-	45 L = 36 h	-	1
Total / Semester: 90 L = 72 h			**4**	**2**

Semester II :

Course	Learning Hours (h) Lectures (L)		Credits	
	Theory	Practical	Theory	Practical
I (USCS 201)	45 L = 36 h	-	2	-
I (USCSP 201)	-	45 L = 36 h	-	1
II (USCS 202)	45 L = 36 h	-	2	-
II (USCSP 202)	-	45 L = 36 h	-	1
Total / Semester	**90 L = 72 h**	**90 L = 72 h**	**4**	**2**
Grand Total / Year	**180 L = 144 h**	**180 L = 144 h**	**8**	**4**

Scheme of Examination:

The performance of the learners shall be evaluated into two parts. The learner's performance shall be assessed by Internal Assessment with 40% marks in the first part & by conducting the Semester End Examinations with 60% marks in the second part.

The Course having Practical training will have Practical Examination for 50 marks at the end of Semester, out of which 30 marks for the Practical task assigned at the time of examination. The 20 marks are allotted as Internal Assessment.

The allocation of marks for the Internal Assessment and Semester End Examinations are as shown below:-

Internal Assessment: It is defined as the assessment of the learners on the basis of continuous evaluation as envisaged in the Credit based system by way of participation of learners in various academic and correlated activities in the given semester of the progamme.

Semester End Assessment : It is defined as the assessment of the learners on the basis of Performance in the semester end Theory/ written/ Practical examination.

Modality of Assessment :
Internal Assessment - 40% 40 marks.

a) Theory 40 marks

Sr No	Evaluation type	Marks
1	Two Assignments/Case study/Project	20
2	One class Test (multiple choice questions objective)	10
3	Active participation in routine class instructional deliveries(case studies/ seminars//presentation)	05
4	Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.	05

b) Practicals 20 marks

Sr No	Evaluation type	Marks
1	Two best practicals	10
2	Journal	05
3	Viva	05

B) External examination - 60 %
Semester End Theory Assessment - 60% 60 marks

i. Duration - These examinations shall be of two hours duration.
ii. Theory question paper pattern :-

1. There shall be four questions each of 15 marks. On each unit there will be one question & fourth one will be based on entire syllabus.
2. All questions shall be compulsory with internal choice within the questions. Each question will be of 20 to 23 marks with options.
3. Questions may be sub divided into sub questions a, b, c, d & e only & the allocation of marks depends on the weightage of the topic.

Practical External Assessment 30 marks