Academic Council 25/05/2011
Item No. 4.91

UNIVERSITY OF MUMBAI

Syllabus for the Computer Programming

Program: Diploma

Course : Diploma in Computer Programming

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

Diploma in Computer Programming Syllabus

(Credit, Grade and Semester System)

To be introduced from the Academic Year 2011 – 2012

The credits earned by the learner in the duration of the one year Diploma programme in Computer Programming are shown in the following Table. Eligibility to this programme is 10+2 or its equivalent. A student can opt for any number of modules (module means theory plus practical of a unit) provided the prerequisites each module have been fulfilled. Credits of this course are transferable to B.Sc. Computer Science. Transfer of credits could be extended to any other subject or discipline subject to the approval of the relevant BOS. Fees to this course would be charged at the rate Rs. 1000/- per module, and Rs. 10000/- for students who will take admission for one year (i.e., all four courses or 12 modules together). Certificate of completion will be awarded to students completing a course of 3 units.

Year	**Sem**	**Course I**		**Course II**		**Course III**		**Course IV**		**Total**
		Th	Pr	Th	Pr	Th	Pr	Th	Pr	
1	**I**	2	1	2	1	-	-	-	-	6
	II	-	-	-	-	2	1	2	1	6
Total		**3**		**3**		**3**		**3**		**12**

Per course per week 1 lecture/period is 48 minutes duration			
	Theory	Practical	Tutorial
Actual contact	3	3	3
Credits	2	1	-

Name of the Programme: Diploma in Computer Programming
(Duration: Two Semesters)

SEMESTER I		
Course Code	**Title**	**Credits**
	THEORY	
DCP 101	**FOUNDATIONS OF PROGRAMMING**	**2 Credits (45 lectures)**
Unit I (No pre-requisite)	Programming logic: Introduction to Scratch, developing Scratch projects for diverse applications Mathematical /algebraic problem solving: Use of flowcharts, introduction to sequential, selection and iterative structures Introduction to imperative programming (IP) using Python	15 lectures 2 3 10
Unit II (No pre-requisite)	Functional programming (FP): What is FP?, Lambda calculus, recursion Problem solving using FP, comparison of FP with IP	15 lectures 6 9
Unit III (Pre-requisite: Unit I / II)	Sorting and searching algorithms: Developing the logic for various search and sort techniques Complexity analysis of an algorithm, comparison of different searching and sorting algorithms	15 lectures 9 6
DCP 102	**FOSS AND SCIENTIFIC COMPUTING**	**2 Credits (45 lectures)**
Unit I (No pre-requisite)	Introduction to Free and Open Source Software (FOSS): What is FOSS? Why and how to contribute to FOSS? Introduction to Linux: Linux distributions, installation Elementary system commands and system administration Introduction to Emacs / vi	15 lectures 2 3 7 3
Unit II	Sage for Mathematical computations: Introduction to Sage Using Sage for computing sets, functions, relations, graphs, matrices, systems of simultaneous equations, counting Computing Minimum spanning tree, shortest path and TSP	15 lectures 2 8 5
Unit III (No pre-requisite)	Statistical computing and Open Office: Overview of probability, distribution functions, correlation and regression, statistical significance LPP, transportation and assignment problem, job scheduling, CPM Introduction to Open Office, Use of spreadsheet applications for statistical computing	15 lectures 4 5 6

DCPPG1	PRACTICAL	2 Credits
	IMPERATIVE AND FUNCTIONAL PROGRAMMING	
Unit I P	Imperative programming practice: Walking-talking cat in Scratch Modify a Scratch project Simulate a calculator in Python Generate / test prime numbers in Python Simulate a sound generator in Python Generate a word-doc vector of a text database	15 periods 2 3 1 2 5 2
Unit II P	Functional programming practice: Compute GCD Generate / test prime numbers Generate Fibonacci sequence Reverse a list Check if palindrome Bubble sort, Quick sort Linear search	15 periods 2 2 1 2 2 2, 2 2
Unit III P	Program analysis practice: Comparison of sorting algorithms: selection sort, bubble sort, insertion sort and quicksort Comparison of searching algorithms: linear search, binary search and hash search	15 periods 8 7
	FOSS AND SCIENTIFIC COMPUTING	
Unit IV P	Practice FOSS tools: Use and contribute to wiki Linux installation (demonstration) Linux commands practice: e.g., login, logout, passwd, init, du; clear, date, df; mkdir, rmdir, cd, mv, cp, rm, ps,; !!, tar, ping, talk, stty, locate, pwd; head, tail, more, grep, man Elementary editing in Emacs / vi: Creation, saving and retrieval of a document, move cursor / screen, type, copy, delete character / word / line; use of search and help	15 periods 2 2 8 3
Unit V P	Practice Sage environment: Set operations Operations on functions and relations Operations on matrices Compute permutations and combinations Sage and / or Python: Solve a system of simultaneous equations Compute minimum spanning tree Compute shortest path	15 periods 1 2 2 1 3 3 3
Unit VI P	Practice Open Office: Writer, Calc, Impress and Draw Develop spreadsheet applications for probability computations, correlation and regression, plot distribution characteristics and analyis Mini project (any one): Develop a software for solving LPP, transportation problem, assignment problem, job sequencing, computing critical path	15 periods 4 5 6

SEMESTER II		
DCP 201	**DATA STRUCTURES**	**2 Credits (45 lectures)**
Unit I PR: 101.1	Data structures and their implementations: Lists, stacks, queues, trees and graphs Introduction to C programming	15 lectures 5 10
Unit II PR: 201.1 and 202.1	Data structures in operating systems: Study/simulate data structures involved in i. Round robin scheduling ii. Priority pre-emptive scheduling iii. Garbage collection	15 lectures 1 5 9
Unit III PR: 201.2	Design of data structures for diverse applications: i. Text Lineification and pattern searching ii. Equation solver iii. Simulation of games such as tic-tac-toe, antakshari, kho-khokho, cricket	15 lectures 5 3 7
DCP 212A	**OPERATING SYSTEMS: A CASE STUDY**	**2 Credits (45 Lectures)**
Unit I PR: 101.1	Introduction and overview of the operating systems: History and evolution of OS Operating systems – functions and structures Information management, process management and resource scheduling, memory management	15 lectures 2 3 9
Unit II PR: 212A.1	System calls, shell commands: What is a system call and how it works, challenges in designing a system call, tracing code of some system calls. For example, sys_exit, sys_fork, sys_read, sys_write, sys_open, sys_close, sys_sched_get_priority_max Introduction to Linux shells, Writing and executing shell scripts. For example, to display date, time, data from a file	15 lectures 10 5
Unit III PR:212A.2	Advanced topics: Linux system security and administration – create, monitor and kill processes, modify process priority, manage file ownership, file access and control to access to the files, create partitions and file systems, mount-unmount, boot-shutdown-reboot, creating and maintaining user accounts, logs, data backup Comparison between Linux and Windows	15 lectures 10 5
DCP 212B	**CREATION AND MAINTENANCE OF E-CONTENT**	**2 Credits (45 Lectures)**
Unit I No Pre-requisite	Information and Communication Technologies: Concept, importance, meaning and nature of Information and Communication Technologies Scope of ICT in Education – Using internet communication facilities like browsers, search engines, e-mail, chat, conferencing, e-databases, blogs, wiki, forums, news, social networks, etc., in teaching-learning, research and publications Challenges in implementing information and communication technologies in education Learning Content Management System: Moodle	 2 8 2 3

Unit II PR: 212B.1	Computer Basics and Content Creation in Moodle: Computer - definition and structure; hardware (keyboard, mouse, scanner, microphone, digital camera, monitor, printer, speaker, screen image projector, hard disk, CD and DVD, mass storage devices); software (OS and application software) Moodle: Installation, introduction to the Moodle operating environment, create a course, organize resources and content development, create a website, collaborative versus individual learning facilities: forum, quiz, blog, etc., database maintenance, grade-book and evaluation support in Moodle	 6 9
Unit III PR: 212B.2	More tools for e-content creation: Introduction to Azure, Shareaza, DC++ for content sharing Introduction to Drupal, Jumla, comparison with Moodle Legal and ethical issues - copyright, hacking netiquette, etc., security and auditing issues Introduction to scripting languages Component content management system, Web content management system	 2 4 2 5 2
USCSPG2	**PRACTICAL**	**2 Credits**
	DATA STRUCTURES	
Unit I P	Programming in C: Hello world, simple calculator Find max / min of a given list Functions factorial (n) / swap(n1,n2) Simulate the following data structures i. List ii. Doubly linked list iii. Stack iv. Two-way stack v. Queue vi. Complete binary tree vii. Binary search tree viii. Precedence graphs	15 periods 1 1 1 2 2 1 1 1 2 1 2
Unit II P	Simulation of OS data structures (in C / Python): Round robin queue Priority pre-emptive queue (2 priority criteria) Grbage collection techniques (2 algorithms)	15 periods 1 6 8
Unit III P	Software development in C / Python: Line beautifier Substring searching Equation solver Mini project (in C / Python): Simulation of data structures involved in a game (For example, tic-tac-toe, antakshari, kho-khokho)	15 periods 2 3 2 8

OPERATING SYSTEM: A CASE STUDY		
Unit IV P	Practice: Linux installation Customization of vi: i. Abbreviations and mapping keys to other keys ii. EXINIT and .exrc iii. Version control system (SCCS, variants) iv. File recovery from temporary copy Comparison of different Linux distributions	15 periods 3 2 4 2 1 3
Unit V P	Understanding source code: Trace, understand and re-document the source code of (any four) system calls Write shell scripts for i. Display time and date ii. List the files / directories that have a given string pattern in filenames / contents iii. Create a file by copying a selected portion (condition given) from an existing file	15 periods 8 1 3 3
Unit VI P	Playing with source code: Modify password criteria Customize copy command Write own file compression Write own text line beautifier Mini project: Study a class of functions related to a task of an OS. For example, user accounts management and security, file and device management, process management and memory management. Trace the source code, re-document it. Compare their functionality in Linux and MS-Windows. Suggest possibilities of modifications and try a few of them. Report your experience.	15 periods 1 1 3 2 8
CREATION AND MAINTENANCE OF E-CONTENT		
Unit IV P, V P and VI P	Project: Create multi-media and interactive e-contents of 10 instructional hours in any subject of your choice and integrate it into an LMS. Make it sharable under FOSS	45 periods

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.)
Following are a few activities that have been suggested with the following objectives:

i. To facilitate the students in making progress through collaborative learning
ii. To enhance their event management skills
iii. To inculcate the value: Help society while you learn more
iv. To promote career building and earning

Activities:

i. Conducting Scratch workshops for junior college and high school students and teachers of disciplines other than CS
ii. Organizing Scratch project contests to model a logical solution for non-mathematical problems (like developing a game or a dialogue, in order to accomplish a goal in an activity)
iii. Test for modeling small problems using the functional approach
iv. Presenting the performance of different algorithms on real life (primary, secondary) and simulated datasets
v. Seminar on “career opportunities for Linux and FOSS users / programmers”
vi. Sage workshop in collaboration with IITB and participation in Sage sprints
vii. Presentation of mini projects
viii. Programming contest
ix. Mini project contest
x. OS utilities demonstration sessions
xi. Quiz on all aspects that have been studied and their relevance in the current context
xii. Workshop / seminar in collaboration with MLUG

Credit Assignment:

Semester I :

Course	Learning Hours(h) Lectures (L)		Credits	
	Theory	Practical	Theory	Practical
I (DCP 101)	45 L = 36 h	-	2	-
I (DCPP 101)	-	45 L = 36 h	-	1
II (DCP 102)	45 L = 36 h	-	2	-
II (DCPP 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 (DCP 201)	45 L = 36 h	-	2	-
I (DCPP 201)	-	45 L = 36 h	-	1
II (DCP 212A / 212B)	45 L = 36 h	-	2	-
II (DCPP 212A / 212B)	-	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 and by conducting the Semester End Examinations with 60% marks in the second part. The allocation of marks for the Internal Assessment and Semester End Examinations are as shown below:-

1. **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.

2. **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 :

A) Internal Assessment - 40%

i. Theory (DCP 101, DCP 102, DCP 201 and DCP 212A / DCP 212B) 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(quiz/ seminars /presentation)	05
4	Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.	05

ii. Practical (DCPPG1 and DCPPG2) 20 marks

Sr No	Evaluation type	Marks
1	Achievements in modeling, programming, demonstration contests	10
2	Practical assignments and Journal /Project	05
3	Viva	05

B) External examination - 60 %

i. Semester End Theory Assessment
(DCP 101, DCP 102, DCP 201 and DCP 212A / DCP 212B) 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 and the fourth one will be based on the 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. Depending upon the distribution of the topics in the Units, there could be sub questions (maximum 5) within a question

ii. Semester End Practical Assessment ((DCPPG1 and DCPPG2) 20 marks

Each practical will be conducted out of 30 marks. A student has to attempt a question (or combination of questions) that comprises of the following types: (i) Trace a given code and document it; write the objective of the code. (ii) Trace the given code in the context of the given objective; fix the bug if any. (iii) Develop a code for a given objective.

Books and References

DCP 101:

1. Scratch 1.4:Beginner's Guide, M. Badger, Packt
2. Head First Python, P. Barry, Shroff
3. Introduction to Functional Programming using Haskell, R. Bird, Prentice Hall
4. Practical Programming: An Introduction to Computer Science (Ch. 1..12 and 14), J. Campbell et al., The Pragmatic Programmers
5. How to Solve it by Computer (Ch. 1..5), R.G. Dromey, Pearson
6. Think Python: How to think like a computer scientist, A. B. Downey, Open source manuscript in pdf format, book published by Cambridge University Press

DCP 102:

1. Data Structures and Algorithms, Aho, Ullman, Hopcroft, Addison-Wesley
2. The C Programming Language, Kernighan and Ritchie, Prentice-Hall
3. http://www.linux-tutorial.info/modules.php?name=MContent&pageid=310

4. *Algorithms + Data structures = Programs, Niklaus Wirth, PHI
5. *Operating Systems: Design and Implementation, A.S. Tanenbaum et al., Prentice-Hall

DCP 201:

1. The Cartoon guide to statistics, L.Gonick, W. Smith et al., Barnes and Noble
2. Discrete Mathematical Structures, Kolman and Busby, Pearson
3. Introduction to Operations Research, Hillier and Lieberman, McGraw-Hill
4. http://fperez.org/talks/0811_baypig_scipy.pdf
5. http://www.sagemath.org/doc/tutorial/
6. http://www.tutorialsforopenoffice.org/

DCP 212A:

1. Operating Systems, Achyut Godbole, MGH
2. Linux Kernel Development, Robert Love, Sams
3. *Modern Operating Systems, Andrew Tanenbaum, Pearson
4. http://www.linuxtraining.co.uk/download/new_linux_course_modules.pdf

DCP 212B

1. Moodle 1.9 E-Learning Course Development: A complete guide to successful learning using Moodle, William Rice, Shroff
2. Moodle 1.9 Teaching Techniques, Susan Smith Nash and William Rice, Packt
3. http://moodle.org/

'*' indicates 'reference books'