

28 MAR 2013

COURSE TEMPLATE

Ref. No.: 20.13.....BCSE.....6957

1.	Department/Centre proposing the course	CSE
2.	Course Title (<i>< 45 characters</i>)	INTRODUCTION TO COMPUTER SCIENCE
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL104
6.	Status (<i>category for program</i>)	UG

7.	Pre-requisites (<i>course no./title</i>)	Basic Knowledge of Computers or CSP 100
----	--	---

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	CSL101/CSL102
8.2	Overlap with any UG/PG course of other Dept./Centre	None
8.3	Supercedes any existing course	CSL101/CSL102

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	☒ Every sem ☐ 1 st sem ☐ 2 nd sem ☐ Either sem
-----	------------------------------	---

11.	Faculty who will teach the course Multiple faculty members may be assigned. This is a first year course.
-----	--

12.	Will the course require any visiting faculty?	NO
-----	--	----

13.	Course objective (<i>about 50 words</i>): Use of high level programming languages for the systematic development of programs. Introduction to the design and implementation of correct, efficient and maintainable programs.
-----	--

14.	Course contents (<i>about 100 words</i>) (<i>Include laboratory/design activities</i>): Concept of an algorithm; termination and correctness. Algorithms to programs: specification, top-down development and stepwise refinement. Iterative versus recursive style; problem solving using a functional style; correctness issues in programming; efficiency issues in programming; time and space measures. Structured style of imperative programming; assertions and loop invariants; programming in an imperative language using advanced features: procedures, functions. Data types, representational invariants. Encapsulation, abstractions, interaction, modularity. Identifying and exploiting inherent concurrency.
-----	--

15. Lecture Outline *(with topics and number of lectures)*

Module no.	Topic	No. of hours
1	- Models of computation. The stored program model. - Mathematical preliminaries - Computer Organization and Basic Principles of Operation - A functional model of computation - The primitive expressions - Function parameter substitution and variable binding - Definition of functions using case analysis	4
2	- Functions as inductively defined computational processes - Recursive processes - Analysis of correctness and efficiency - Tail-recursion and iterative processes - Correctness of an iterative process - Examples of iterative processes	4
3	- Analysis of correctness and efficiency - Correctness - Efficiency - Efficiency, Why and How? - In the long run: Asymptotic analysis and Orders of growth - More examples of recursive algorithms	4
4	The Imperative model of computation - The primitives for the imperative model - Variables and the assignment instruction - Assertions - Conditional instruction and iterative instructions (for, while) - Functions and procedures in the imperative models - Step-wise refinement and Procedural Abstraction	5
5	Linear Data Structures: - Lists, Arrays, Strings - Sequence Operations - Stacks and queues - Searching & Sorting - Branching Data Structures & Binary Search Trees, - Hashes & Multilevel queues	8
6	Executable specifications and rapid-prototyping Examples of step-wise refinement	6
7	Encapsulation, abstractions, interaction, modularity and objects.	6
8	Identifying and exploiting inherent concurrency. Concurrency and Multithreaded computations	5
9		
10		
11		
12		
COURSE TOTAL (14 times 'L')		42

16. Brief description of tutorial activities

NA

17. Brief description of laboratory activities

Module no.	Experiment description	No. of hours
1	Multiple Programming Assignments of varying difficulty will be given on topics such as	4
2	-- Numerical Computations	4
3	-- Working with Linear data structures -- lists and strings	4
4	-- Searching & Sorting	4
5	-- Multi-objective search	6
6	-- Concurrent processes --- a sample multithreaded program	6
7		
8		
9		
10		
COURSE TOTAL (14 times 'P')		28

18. Suggested texts and reference materials

STYLE: Author name and initials, Title, Edition, Publisher, Year.

- 1, Subashish Banerjee, Lecture Notes in Computer Science, 2011
2. Python Programming: An Introduction to Computer Science by John Zelle. or suitable language reference depending on Language used for instruction (Python/SML/OCAML/Haskell/etc)

19. Resources required for the course (*itemized & student access requirements, if any*)

19.1	Software	Python, OCAML Interpreters
19.2	Hardware	Computer Terminals for All Students
19.3	Teaching aides (videos, etc.)	
19.4	Laboratory	Computer Centre/First Year Teaching Lab
19.5	Equipment	
19.6	Classroom infrastructure	Classrooms for students
19.7	Site visits	None

20. Design content of the course (*Percent of student time with examples, if possible*)

20.1	Design-type problems	20%
20.2	Open-ended problems	
20.3	Project-type activity	
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date: Mar 24, 2013

(Signature of the Head of the Department)