[A-3]COURSE SYLLABUS

1. Course Description (Approximately 100 words)

This course aims to present the major ideas from the area of formal semantics of programming languages AND demonstrate their importance to language design and implementation, and to program verification, specification and refinement. Students develop simple mathematical techniques for describing the semantics of programs and reasoning about their denotations and are given an overview of the relationships between program semantics and logics of computation.

2. Course Goals and Objectives (Approximately 100 words)

By the end of the module, students will:
1. understand the concept of formal semantics of programs and the mathematical techniques used to describe the semantics of programming languages;
2. be able to use these in reasoning about programs and computations;
3. have a deeper understanding of the notions o computation and data structure;
4. have a deeper understanding of program specification,verification and refinement, as well as the issues involved in integrating software components written in different languages.

3. Textbook (Title, Author, Publisher, Year of Publication, etc.)

Joseph A. Goguen and Grant Malcolm. Algebraic Semantics of Imperative Programs. MIT Press, 1996.

4. Reference

Joseph A. Goguen and Grant Malcolm. Algebraic Semantics of Imperative Programs. MIT Press, 1996.

5. Course Requirements and Grades

Exam 70%
Continuous Assessment Test (CAT) 20%
Assignment 10%

6. Course Calendar

Week	Main Content
1	Lecture 1: syntax and semantics
2	Lecture 2: informal semantics
3	Lecture 3: programming languages and compositional semantics
4	Lecture 4: logic and computation
5	Lecture 5: Denotational Semantics: programs as functions
6	Lecture 6: Denotational Semantics: loops and recursion
7	Lecture 7: Denotational Semantics: fixed points
8	Lecture 8: Semantics of Data Structures: algebraic specifications
9	Lecture 9: Semantics of Data Structures: ; models
10	Lecture 10: Semantics: equational deduction & computation as proof
11	Lecture 11: Axiomatic Semantics: specification of programs and languages
12	Lecture 12: Axiomatic Semantics: correctness proofs
13	Lecture 13:Axiomatic Semantics: objects, data structures and local states
14	Lecture 14: Overview and revision