DBLN CPSI 4490

This course is being offered at Griffith College, CAPA's academic partner in Dublin. The Irish academic system differs from the US, particularly with grading. Griffith College professors expect students to undertake a good deal of independent study to achieve a high mark in their classes. For additional information about this class, please contact the Boston Program Advising Team at 1-800-793-0334.

Introduction to Formal Design Methods

Continuous Assessment: 50%
Exam: 50%

Intended Module Learning Outcomes

On successful completion of this module learners will be able to:

1. use assertions to prove the axiomatic semantics of a simple programming language
2. use pre/post conditions to prove the correctness of simple programs
3. prove the correctness of programs using loops, loops within loops and sequential loops

4. derive programs from initial specifications using different methods of construction based on axiomatic semantics.

Module Objectives

The key objectives of this module is are to teach learners to understand and apply those branches of logic necessary to develop correct programs; to develop an understanding of the mathematics required to develop programs from initial specifications through to implementation by using formal techniques; how to prove the correctness of a given program based on its pre and post conditions; apply a formal semantic model to the derivation of programs; methods necessary to derive correct programs given a formal specification in terms of pre/post conditions. This module prvoides learners with an understanding of how to apply mathematical reasoning to program development based on state based systems. As such it links to both the Programming Paradigms module running concurrently with it and also with the 4th year module on Formal Specification.

Module Curriculum

Introduction

- Explain the need for formal methods in the construction of programs;
- Examples of programs with bugs;
- Writing assertions over sequences;
- The use of predicates to describe states in programs.

Axiomatic semantics of a guarded command language

- Predicates as assertions in programming languages;
- The role of assertions in the execution of programs. The semantics {P} S {Q};
- Definition of skip; assignment; if .. fi; do .. od;
- Definition of an invariant and its use in proving the correctness of loops;
- Proving correctness using definitions;
- Proving correctness of complete programs using assertions and axiomatic semantics;

Formal derivation of programs

- Writing specifications of problems using pre conditions and post conditions;
- Formally deriving programs from initial specifications to complete program code;
- Method 1: Problems of the form {P} do b® S1 {P}od {P ÙQ}
- Method 2: Replacing a constant with a variable;
- Problem domains involving loops within loops;
- Method 3: Strengthening an invariant;
- Invariant diagrams;
- Applying formal approach to searching and sorting;
- Searching for optimal solutions – O(logN) and O(N) solutions to computational problems.