NEW ZEALAND INSTITUTES OF TECHNOLOGY AND POLYTECHNIC QUALIFICATIONS IN INFORMATION & COMMUNICATIONS TECHNOLOGY

PRESCRIPTION: PP400 PROGRAMMING PRINCIPLES

AIM OF MODULE:	The student will develop an understanding of elements of programming and the disciplines of good programming technique. The student will acquire basic competence in a selected programming language and will apply this language to simple tasks using good programming techniques.
CREDITS:	7
STUDENT LEARNING HOURS:	70
CONTENT REVISED:	Pre 1996
PRESCRIPTION EXPIRY DATE:	November 2013
NOTE:	THIS IS A COMPULSORY DIPICT L5 MODULE

Level and Assessment Schedule

	Highest Skill Level				Suggested Assessment Percentage
TOPICS	R	C	A	P	
1. Procedure Design			*		40
2. Programming Features			*		5
3. Language Syntax	*				5
4. Translating Procedures into Program			*		10
5. Writing, Compiling, Testing, and Modifying Programs			*		25
6. Program Documentation			*		15
					100

LEARNING OUTCOMES

The student will:

A 1 Demonstrate appropriate use of sequence, selection and iteration to design well structured procedures to carry out simple tasks.

A 2 Demonstrate the use of basic programming features in the selected programming language.

R 3 Describe the syntax of a subset of the selected programming language.

A 4 Translate simple procedures from an appropriate depiction of logic into the selected programming language.

A 5 Design, write and compile simple programs in the selected language, resolve syntax errors, test the programs, and modify the programs as required to meet the original procedure design objectives.

A 6 Document programs.

CONTENT

NOTE: "Procedures" refers to procedures, functions, subroutines, methods, or the like.

1 Suitable logic depiction methods such as structure diagrams, structured English, and UML may be used.

➢ Procedures need not relate to business data processing but the selection of tasks should introduce design techniques which will be required for typical data processing problems.

➢ A *maximum* of three levels of nesting of selection/iteration is recommended.

➢ Examples of procedures to indicate the level of difficulty:

- Calculation of age, given birth year
- Conversion of temperatures from Celsius to Fahrenheit
- Calculation of sum and average, output of highest and lowest of a list of numbers
- Print out a list of numbers in reverse order
- Merge two ordered lists of numbers

2 Programming features such as input, output, operations (eg. arithmetic) on data, data storage and movement, use of procedures should be included.

- The action of individual items of syntax in the selected programming language should be explained. This includes information about any data elements it works with, produces or changes.

- The subset will include supplied procedures, logic control, assignment and comparison operators that are required to write programs for this module.

2-5 THE LANGUAGE

- Examples of suitable languages:

  - BASIC or Visual Basic
  - Pascal or Delphi
  - COBOL
  - C or C++
  - Java
  - Jade

- An integrated programming environment using either a procedural or object oriented language is recommended.

6 Program Documentation includes internal and external documentation such as author, purpose of program, date written, modifications, variable names, procedure names, logic plan, testing etc.

NOTES AND COMMENTS

- The intention of this module is to get the student to implement problem solutions they have completed in PD500, not to produce commercial programs.
- The module requires students to acquire skills in taking the fundamental building blocks of procedure design (selection, iteration and sequence) and applying these to simple problems. The vehicle for testing the resulting procedures will be the selected programming language. The major part of the assessment is on procedure design rather than on proficiency in the selected language. Other modules concentrate on the learning of programming languages.
- Learning a first programming language is, for many students, their first encounter with extensive application of logic. Often their performance in this activity falls short of their expectations, which will be based on previous success at knowledge-based modules. The training provider should recognise and address this factor or risk the student being alienated from programming.
- Some students respond well to learning the syntax of the language first from simple examples, and then learning the basics of procedure design. Other students prefer to learn the theory of procedure design and then to apply this theory while acquiring knowledge of the syntax of the selected language.
- Section 6 (Document programs) is moved from PD500 so that it is done while students are using a programming language. The weighting for Procedure Design is reduced to compensate because so much of its content is covered in PD500.