CHAPTER 1

Introduction

It is difficult not to agree with Clarke's (1992, 6) proposition that specialist information technology (IT) courses in secondary schools should 'enthuse and skill young people to the point where they become designers/developers of IT, instead of simply being consumers/users of IT'. A similar point of view is expressed by Newlands and Teague (1993, 15) in discussions about an appropriate programming language for first year tertiary students. These authors comment that secondary students should be involved in 'developmental work which will provide a more exciting and stimulating course for enthusiastic students'. It therefore appears an appropriate investigation to ask if students may gain benefits by developing applications within an object-oriented programming environment as opposed to the established procedural programming environment. Entwined with this investigation is the possibility that the addition of sophisticated user interfaces to applications will enthuse students and provide more stimulating and exciting developmental work.

The C and Pascal programming languages, among others, have recently been packaged with object-oriented capacity and object libraries. The advent of these languages, incorporating both procedural programming and object-oriented programming capability, at a reasonable price for schools raises the question of whether an exposure of senior students to object-oriented programming is a beneficial experience. The examination of whether object-oriented programming and sophisticated user interfaces have educational benefits for secondary school students is firstly dependent upon the dictates of the information technology curriculum. Within Queensland, influencing factors are the syllabus, the programming language specified by the syllabus, the requirements of tertiary institutions and the school.

Information Processing and Technology (IPT) is a Board of Senior Secondary School Studies (BSSSS) approved course of study for Year 11 and 12 students in Queensland. The **syllabus** for IPT contains the following statement (BSSSS 1991, 1):

Information Processing and Technology is a complex intellectual discipline which deals with information systems, algorithms and programming, artificial intelligence, computer systems and the social and ethical implications of information technology. Its emphasis is software development rather than the use of application packages.

Within the Algorithms and Programming topic, the syllabus (BSSSS 1991, 12) states that the aim is to cultivate software development expertise and programming skills in students, with a focus on the design and implementation of algorithms for the solution of practical problems. The syllabus, while avoiding a clear statement about which type of programming language should be used to support the aims of the topic, makes a number of references to implementing algorithms through 'at least one' procedural language, and to the production of structured, modular programs (BSSSS 1991, 14). Specific Algorithms and Programming objectives emanating from the syllabus are listed in Appendix A and provide a further understanding of the general thrust of the Queensland syllabus.

A 'snapshot' of information technology courses leading to tertiary entrance throughout the Australian states indicates that all such courses include a significant amount of algorithms/programming, with most courses emphasising the use of procedural languages (Clarke 1992, 4). The majority of those **schools** in Queensland which offer the course Information Processing and Technology (IPT) choose to support the algorithms and programming topic with procedural programming using TurboPascal. A survey of schools offering Information Processing and Technology indicated that sixty of the seventy-one respondents (56% response rate) use procedural TurboPascal as the main language for the Algorithms and Programming topic (King, Feltham and Nucifora 1994, 21). The syllabus provides a time allocation for the Algorithms and Programming topic of between twenty percent and thirty-five percent of the course, a prescriptive limit which is roughly equivalent to between fourteen and twenty-five weeks (BSSSS 1991, 5).

Newlands and Teague (1993, 15) consider that while there are many reasons for **tertiary institutions** to move from Pascal to C as a first programming language, there are very strong arguments for secondary schools to use a well-structured procedural language such as Pascal. Searle (1985) describes Pascal as a language very suited for use by senior secondary computer science students.

Object-oriented programming is an evolutionary step in the development of programming techniques. This evolution has been stimulated by the need for programming languages to cater for more features in graphical user interfaces, greater complexity in applications, together with improvements in the way people think and communicate with a programming language. Weiskamp, Flamig and Heiny (1991, 2) illustrate the evolution of programming languages by describing three ages. The three ages are depicted in Figure 1.

Age of chaos 1950s to 1960s

jumps go to unstructured variables variables scattered throughout the program Age of structure 1970s to 1980s

if then else while loops blocks records units Age of objects 1990s to ?

objects messages methods inheritance

Figure 1. The evolution of programming languages.

An object is defined by Weiskamp, Flamig and Heiny (1991, 3) as 'a language construct that ties data with the procedures and functions that operate on the data'. The tying of data and code into an object enables the use of objects as building blocks for more complex objects or applications.

Borland (1990b, 73) describes three properties of an object-oriented programming language as being encapsulation, inheritance and polymorphism. Encapsulation is the construct of combining data and associated procedures and functions together to form an object data type, effectively placing a protective barrier around each piece of data and protecting the data from unintended use (Cox 1986, 8; Borland 1990b, 73; Martin 1993, 19). Inheritance involves the construction of a hierarchy of object classes, each descendant having access to each of its ancestor's code and data, thus providing a means of automatically broadcasting code to appropriate parts of an application (Cox 1986,8; Borland 1990b, 73). Polymorphism refers to the ability of an object to share an action with its hierarchy but being able to use the action as it wishes (Borland 1990b, 73).

The three described properties of object-oriented programming establish a foundation for a number of powerful benefits such as modularity, reliability, and reusability (Borland 1990b, 73; Giovanni 1992, 130). Rubenking (1992, 157) describes the benefits of object-oriented programming as going beyond the accepted encapsulation, inheritance, and polymorphism to a higher level of modularity, reliability, and reusability than is possible with conventional procedural programming.

One potential of object-oriented programming systems and their libraries is that a student may 'inherit' a complicated class hierarchy to provide a standard user interface. A student may potentially be able to couple application logic with a user interface skeleton to produce a sophisticated application - an application which can handle multiple resizeable windows, mouse support, menus, hotkeys

and icons and have, for example, a distinctly 'Turbo' look-and-feel (O'Brien 1991, 20). As indicated by Borland (1990a, 1), 'With Turbo Vision and objectoriented programming, you don't have to re-invent the wheel - you can inherit ours!'. Figure 2 illustrates a rudimentary comparison between the normally amateurish interface which a student experiences and the inherited interface of an object library.



Figure 2 A 'Hello World' comparison.

The gradual movement of the windows environment into computing classrooms, presently restricted only by issues of cost, will also increase the need and consideration for object-oriented programming. The evolving complexity of software interfaces, such as the windows environment, establishes object-oriented programming as a necessity rather than an option (Cox 1986, 28; Martin 1993, 18). It may be assumed that, while little object-oriented programming is being implemented, a large number of schools are watching the development of object-oriented programming with interest.

This examination of the context of object-oriented programming for secondary schools suggests that procedural programming has a firm position in information

technology courses in secondary schools. Object-oriented programming appears to have a supplementary place in schools by allowing students to 'inherit' an industry standard user interface. This possible supplementary position is sustained by the addition of object-oriented programming capability to procedural languages such as Pascal and C and by the possibility that attaching a sophisticated user interface to applications will enthuse students.

Theoretical framework

Learning style

Lewin (1951) outlined three interacting factors which influence the learning process: the student, the instructor, and the school environment. Bloom (1976) subsequently proposed a theory of school learning which attempts to explain variation in student achievement on a learning task in terms of three interdependent variables: cognitive entry characteristics, affective entry characteristics, and quality of instruction. Figure 3 indicates the major variables in Bloom's theory of school learning.



Figure 3. The major variables in Bloom's theory of school learning.

Cognitive entry behaviours are the knowledge, skills, and competencies which are essential prerequisites for the learning to be undertaken (Bloom 1976, 31). It

is estimated that cognitive entry behaviour may account for one half of the variance in cognitive educational achievement, with sufficient evidence to suggest that a causal link exists between cognitive entry behaviour and cognitive achievement (Bloom 1976, 68).

Affective entry characteristics are the extent to which the student is or can be motivated to engage in the learning process. Bloom (1976, 104) states that there is a clear relation between affective characteristics and related measures of school achievement. It was estimated that affective entry characteristics may account for one-fourth of the variance on cognitive achievement with a suggestion of a causal link as well as a predictive component.

Quality of instruction is the extent to which the instruction to be provided is appropriate to the learner (Keefe 1987, 4). Bloom (1976, 115) considered that the quality of instruction was determined more by the teaching method and learning environment than by teacher characteristics or physical characteristics of the classroom. The quality of instruction appeared to account for one-quarter of cognitive achievement (Bloom 1976, 135).

The power of Bloom's theory is emphasised by the assertion that the combination of cognitive entry behaviours, the quality of instruction, and affective entry characteristics account for more than eighty percent of variation of cognitive educational achievement (Bloom 1976, 174).

Keefe (1987, 5) contends that, while Bloom's theory provides a useful framework for understanding school learning, there is a need for a framework which allows for a greater range of diagnostic information. Diagnostic information is required which would provide direction in catering for the wide variety of student differences and in making decisions about the use of appropriate teaching approaches.

Letteri (1982) proposes a learning style model, which is based on an information processing perspective of learning which can be traced from information-processing theory and brain physiology (Santostefano 1978). Figure 4 presents the main factors in Letteri's model.



Figure 4 Information Processing, General Operations Charles A. Letteri, 1982.

Letteri proposes that learning involves the application of operations which act on new information so that it becomes a part of long-term memory. Cognitive control, the ability to control the information processing operations, is initially involved in perceptual reception, raw data being received through the eye, ear, nose, tongue and skin networks. Perceptual memory maintains this data for a brief period of time enabling decisions to be made about whether the information is previously known. The filter phase involves decisions about whether the information should be rejected, memorised, transformed or learnt. Short-term memory maintains small amounts of information for brief periods of time and thus allows working memory to process the information. The processes and functions performed by working memory brings short-term memory and the cognitive operations of long-term memory together to permit learning. Everything learned is held in long-term memory in an accessible and organised fashion.

Letteri (1988, 33) asserts that cognitive controls are skills (analytic, spatial, discrimination, categorisation, sequential processing, memory) that students can be trained to use to control the operations of their information processing systems. The combination of the learner's cognitive control with other developmental, psychological, and environmental preferences constitutes learning style (Letteri 1988,23).

Essentially, learning style is said to be composed of individualised cognitive, affective, and environmental factors which provide relatively stable indicators of how learners perceive, interact with, and respond to the learning environment (Keefe, Monk, Letteri, Languis, and Dunn 1989). A descriptive statement about learning style is proposed by Smith (1982, 23)

What do we mean by style? It has long been apparent to teachers, educators, and observers that people differ in how they go about certain activities associated with learning. They differ as to how they go about 'information processing', or putting information through their minds. Some people like to 'get the big picture' of a subject first then build to a full understanding of that picture by details and examples. Other people like to begin with examples and details and work through to some kind of meaningful construct or way of looking at an area of knowledge out of these details. Some like theory before going into practice. Others don't.

Cognitive styles are 'information processing habits representing the learner's typical mode of perceiving, thinking, problem solving, and remembering' (Messick 1976). Within Letteri's model, cognitive styles are preferred ways of perceiving, filtering and retaining information that are characteristic of a learner. *Affective styles* encompass 'those aspects of personality that have to do with attention, emotion, and valuing' (Keefe 1987, 9). Affective styles influence decisions

made by a learner about what to do with given information and the amount of effort expended in memorising information. *Physiological styles* describe 'the characteristic learning-related behaviours of the human body. Physiological styles are biologically-based modes of response that are founded on sex-related differences, personal nutrition and health, and accustomed reaction to the physical environment' (Keefe 1987, 13). Because the decision to learn requires a concerted effort by the learner to process information (Letteri 1987, 30), students with physiological problems will have difficulty expending the required concentration.

Keefe (1987, 41) contends that learning style is capable of providing a deeper and more profound view of the learner than that provided by Bloom, and that learning style is a basic framework upon which a theory and practice of instruction can be built. A number of researchers have assumed that learning style is measurable and that the instruments do provide valid and reliable measures of the learning style construct (Cross 1976; Keefe 1989).

Motivation

Letteri's learning style model provides a suitable framework for gaining diagnostic information about cognitive entry behaviours within Bloom's theory of school learning. There is a need to pursue diagnostic information within the realm of affective entry characteristics. Recent theory and research on motivation suggests that an appropriate approach is based upon a model of 'expectancy times value' (Feather, 1982). This theory proposes that student motivation is a product of the expectation that students will succeed if they apply themselves to a task, and the degree to which the student values the task (Good and Brophy 1991, 278). The product theory assumes that there will be no motivation if either student value of the task or student expectancy of success is missing. The implication of this theory for teachers is that teachers should ensure students can have a reasonable expectation of achieving success

in an activity if they apply reasonable effort, and should help students appreciate the value of the activity. Figure 5 represents a conceptual framework of the theory constructed from Good and Brophy's (1991, 279-298) dissection of expectancy times value theory.

	Extrinsic value Rewards can act as incentives for good performance. The application value of a task can help
	appreciation of learning the task.
Realising that there is a direct relationship between amount of effort expended and	Opportunity to compete can add excitement to learning.
expected level of success.	Intrinsic value
Realising that the potential for success lies with internal factors rather than with	Students prefer tasks which allow them to actively respond.
external factors. Realising that success can be achieved	Students value tasks which include higher-level objectives and divergent questions.
through self rather than believing that fate is determined by external control.	Students enjoy tasks from which they get immediate feedback.
Realising that one has sufficient ability to achieve success if sufficient effort is applied.	Students like tasks which include gamelike features.
Realising that failure is due to insufficient effort or confusion, not due to lack of chility or upcontrolloble factors	Students value tasks which allow them to create finished products.
Realising that academic ability is	A Students enjoy tasks which include fantasy or simulation elements.
continually increased through persistent learning rather than being a predetermined limit.	Students enjoy tasks which allow them to interact with their peers.

Figure 5. Inferences of the expectancy X value motivational theory.

The Research problem

This study will investigate whether object-oriented programming has educational value for secondary school students. An analysis of the requirements of senior secondary student programming and the brief literature review in the introduction

suggests that using object-oriented programming as a supplement to procedural programming would serve as a suitable pedagogical development.

The study will also consider the possibility that the addition of sophisticated user interfaces will enthuse and stimulate students. It is expected that the inherited user interface will add to motivational value as a consequence of its application value, the opportunity for active response, inclusion of higher level objectives, immediate feedback, and the opportunity to produce a finished product. Biggs and Moore (1993, 259) believe that the value of the student task may be enhanced by emphasising the quality of the product.

Establishing answers to the general research questions is then dependent upon a set of subsidiary questions involving alternative programming environments and the cognitive skill factors discussed in relation to learning styles and motivation:

Is there a difference, in student achievement, obtained by adding a sophisticated user interface within a procedural programming environment or within an object-oriented environment?

Is there a relationship between any of the cognitive skill elements and student achievement within an object-oriented programming environment or student achievement within a procedural programming environment?

Are the cognitive skill elements which may be useful in predicting student achievement in procedural programming different to the cognitive skill elements which may be useful in predicting student achievement in objectoriented programming?

Will the addition of a sophisticated user interface to students' programming applications add value to the learning and thus make the course more enjoyable and intrinsically rewarding for students?

Significance of the study

Van Merrienboer (1988, 185) believes that research on cognitive styles and programming achievement has the potential to provide a model of relationships between these two variables. The establishment of a model would permit the design of instructional strategies and materials so that the match between instruction and student cognitive styles are optimised. Such a model would allow the improvement of student academic success because learning problems are more frequently related to the type and level of the cognitive processes required to learn the material rather than to the difficulty of the subject matter (Letteri 1988, 22). A model of the relationships between student cognitive skills and programming achievement would also permit direct intervention in the learning process to provide students with relevant cognitive skills training.

The inexpensive packaging of object-oriented programming, object libraries and procedural programming combined with the commercial success of object-oriented software development, has presented strong questions about whether students should begin experiencing object-oriented programming. If this study indicates that students in secondary schools may benefit from the use of object-oriented programming then there obviously exists a strong case for introducing object-oriented programming into schools.

Wiersma (1991, 370) considers that 'one contribution of a study is the identification of questions for further research.' If this study indicates that students in secondary schools may benefit from the use of object-oriented programming then there is a case for further investigation. For example, would object-oriented programming improve the problem-solving skills of students; will the study of both procedural and object-oriented programming enhance the transfer of problem-solving skills to other domains; does object-oriented programming provide an extra dimension for problem-solving; does object-oriented programming cater for a wider range of learning styles?

CHAPTER 2

Review of the related literature

The research domain

Bordens and Abbott (1991, 463) state that 'The heart of the research process is identifying important variables to study, measuring those variables,'. This chapter discusses some previous studies of object-oriented programming in educational settings and identifies relevant variables which may be operationally defined within the context of the research question. Learning Style, a source of diagnostic information identified within the theoretical framework of this study, is one of the more relevant variables.

Object-oriented programming

Reports of research into the use of object-oriented programming within the classroom is limited and a search of Educational Resources Information (ERIC), Psychological Abstracts and the Educational Index using keywords such as object and (programming or programing) produced very few titles.

Significant reference to the research question involves the description by Borne and Girardot (1991, 93-98) of the use of Smalltalk-80, a French programming language, to teach primary school children to program effectively using object-oriented concepts. Borne and Girardot (1991) propose that objectoriented programming is teachable to thirteen year old children, is well suited for the development of user interfaces, and has a part to play in learning about computers because there is a close correspondence between a real world object and a data processing object.

Some implied support for the Borne and Girardot proposals is found in the educational literature. Use of object-oriented programming to inherit user interfaces within an educational context is provided by Cunningham, Corbett, and Bonar (1987), Milet and Harvey (1989) and Greenberg (1991, 267-73). Rosson and Alpert (1988) conclude that object-oriented design has cognitive implications for the flexibility and management of problem decomposition, and that it may result in better designs. Goldberg and Kay (1977) also suggests that object-oriented programming is teachable to young children.

The use of object-oriented concepts which have some relationship to the current research question has been explored in a variety of environments. Boxer (diSessa and Abelson 1986) presents computational objects in the forms of boxes and allows an easy adaptation to object-oriented programming. Although showing great promise, the Boxer environment remains essentially experimental. Rehearsal (Gould and Finzer 1984; Finzer and Gould 1987) allows graphic programming in the form of a theatre metaphor, where the basic components are performers and the performers interact with each other by sending cues. In the Rehearsal environment children do not learn how to program in the traditional sense of computer programming. Playground (Fenton and Beck 1989) is an object-oriented environment where children construct simulations by providing graphical objects with laws. Milet and Harvey (1989) suggest that objectoriented concepts has some counterparts in hypermedia systems. The authors suggest that developing hypermedia applications using Hypertalk, the programming environment for Hypercard applications in Apple Macintosh, can help in the understanding of object-oriented programming and vice versa.

Complementing a procedural programming course with the use of an objectoriented language receives some support from Lawson (1985a, 541; 1985b, 542). Lawson contends that a single programming language, or one class of programming languages, restricts the problem solving approach. Even at the introductory computer education level, Lawson believes that utilising a variety of procedural and object-oriented programming languages will enhance students' views of the use of computer systems for problem solving. McGrath (1988, 467-484) supports the view that the use of two programming languages has benefits for students.

Turkle and Papert (1992, 3-33) propose that object-oriented programming has created an opening for epistemological pluralism, an opportunity to analyse the different ways in which people acquire knowledge. However, the implication that object-oriented programming provides alternatives to traditional analytical programming knowledge acquisition is more relevant within a context of visual programming than the object-oriented programming context of the research question.

Learning style

Learning style has considerable relevance to the research question by providing diagnostic information within the umbrella of Bloom's theory of school learning (Keefe 1987, 41). The learning style framework is supported by lucid operational definitions and a considerable number of diagnostic research tools. A number of researchers have assumed that learning style is measurable and that the instruments do provide valid and reliable measures of the learning style construct (Cross 1976; Keefe 1989).

Definitions of the learning style construct, while having some variation in emphasis do have common elements. Essentially, learning style is said to be composed of individualised cognitive, affective and environmental factors which provide a relatively stable indicator of how a learner perceives, interacts with, and responds to the learning environment (Claxton and Ralston 1978; Smith 1982; Keefe, Monk, Letteri, Languis, and Dunn 1989).

The affective component includes the amount of structure and authority the learner prefers, expectations and motivation, and the degree of interest in the subject matter to be learned (Smith 1982). The environmental component of learning style can range from the very specific things such as preferred room temperature to the amount of emotional support learners need in the immediate learning environment (Smith 1982).

Considerable emphasis is placed upon cognitive style as a factor of learning style and this emphasis is supported by the huge breadth of cognitive style research. Cognitive style theory (Ausburn and Ausburn, 1978; Messick, 1984; Kolb, 1984) suggests that individuals develop a preferred way of thinking, problem solving, and interacting with the environment. Cognitive style refers to the manner in which individuals process information (Witkin, Moore, Goodenough, and Cox, 1977). Richter (1992, 19) defines cognitive style as being part of personality organisation and representing a characteristic mode of information processing which involves a myriad of metaprocesses. Cognitive styles, then, are stable individual preferences for perceptually organising and conceptualising the environment. Bishop-Clark (1992, 3) makes the point that style measures preferences and theoretically it is value free, implying that one cognitive style is no better than another cognitive style; it is simply different.

Cognitive factors of learning style (Smith 1982) include field-independence versus field-dependence (Witkin and Goodenough, 1981); conceptualising and categorising (Kagan and Kogan 1970; Kolb and Fry 1975; Messick 1984); reflectivity versus impulsivity as measured by the Matching Familiar Figures Test (O'Donnell, Paulson and McGann, 1978).

A number of scales and instruments have been developed to measure individual differences in learning style (Myers 1962; Canfield and Lafferty 1974; Kolb 1976; Gregorc 1984; Keefe, Monk, Letteri, Languis, and Dunn 1989). The intention is generally to provide a single instrument which assesses a broad spectrum of research-based learning style elements composed of mainly cognitive, affective, and physiological/environmental domains (Keefe, Monk, Letteri, Languis, and Dunn 1989). Table 1 lists some of the more popular learning style instruments.

Programming is a complex activity and requires a variety of skills (Shneiderman, 1980; Pea, Kurland and Midian 1983). The exploration of the relationship between learning style, especially cognitive styles, and programming achievement has attracted a number of studies (Fletcher, 1984; Webb, 1984, 1985; Bradley, 1985; Pommershein, 1986; McCoy and Burton, 1987). Since computer programming requires a combination of many skills, examining the relationship between a specific cognitive factor and several programming component skills may impact upon programming instruction. However, only a few studies have examined specific cognitive predictors with regard to programming component skills (Foreman 1988, 6).

Instrument	Definition of Learning Style	Instrument	Applications/Implications
Myers-Briggs Type Indicator	Learners are orderly and consistent in the way that they use perception and Judgement. Perception includes the processes of becoming aware of things, people or ideas. Judgement includes the processes of coming to conclusions about what has been perceived. An individual's type can be measured along four bipolar dimensions: extroversion/introversion; sensing/intuition; thinking/feeling and judgement/perception.	A forced-choice, self-report personality inventory which consists of 126 items yielding four scale scores. It is essentially for use with adults and can be administered individually and in groups. (50 minutes administration).	Adults may find the type concepts useful for helping to understand basic preferences for learning which can assist in determining compatibility between learning type, method of instruction and other personal or environmental influences on learning.
Canfield Learning Styles Inventory	Individual learning style is derived from: (a) academic conditions (relations with instructor and peers); (b) structural conditions (organisation and detail); (c) achievement conditions (goal setting, competition); (d) content (numbers, words etc); (e) mode of preferred learning (listening, reading, iconic and direct experience; and (f) expectation of performance level (superior through satisfactory).	A self-report instrument based on rank ordering of choices for each of 30 questions. For use with junior high through adult levels. (15 minutes administration)	Its major use is to develop instructional materials for whole classes or individual students. The LSI is considered a tool to aid in understanding student's difficulties in completing academic units and for counselling. Emphasis is placed on attitudinal and affective dimensions and the Inventory focuses on such applications
Gregorc Style Delineator	Learning style consists of distinctive, observable behaviours that provide clues to the functioning of people's minds and how they relate to the world. These 'mind' qualities suggest that people learn in combinations of dualities: (a) concrete- sequential; (b) concrete-random; (c) abstract-sequential; and/or (d) abstract- random. Preferences for a particular set constitute a learning style.	A self-report instrument based on a rank ordering of four words in each of 10 sets. Observation and interviews suggested that these words can be used to aid in categorising learning preference patterns or modes. For use with upper junior high students through adults. (15 minutes administration)	
Kolb Learning Style Inventory	Learning style is a result of hereditary equipment, past experience, and the demands of the present environment combining to produce individual orientations that give differential emphasis to the four basic learning modes postulated in experiential learning theory: Concrete Experience (CE); Reflective Observation (RO); Abstract Conceptualisation (AC); and Active Experimentation (AE).	A self-report instrument based on a rank ordering of four possible words in each of nine different sets. Each word represent one of four learning modes: feeling (CE); watching (RO); thinking (AC); doing (AE). (5-10 minutes administration)	Emphasis is placed on individual awareness of personal learning style and available alternative modes. Knowledge of learning style differences should encourage the design of instructional experiences and to enhance individual strengths and develop non-dominant orientation.
NASSP Learning Style Profile	Learning style is a composite of those elements that serve as a relatively stable indicator of how a learner perceives, interacts with, and responds to the learning environment. The learning style elements are classified into cognitive, affective, and physiological/environmental domains.	A forced-choice, self-report inventory which consists of 126 items. It is essential for use with students from the sixth to twelfth grades. (40 minutes administration.)	A diagnosis tool providing the basis for a more personalised approach to student advisement and placement, to instructional strategy, and to the evaluation of learning

Table 1.Some popular learning style instruments. (The information and format of
this table is adapted in part from Dunn, DeBello, Brennan and Murrain
(1981) and Sewall (1986)).

A causal relationship between learning style and programming skills hasn't been established (Linn 1985) although there appears to be considerable evidence

suggesting that motivation and achievement increase when the instruction is matched to the learner's preferred learning modes (Dunn 1980; Jenkins 1982; Pizzo 1982; Curry 1983; Dunn 1984; Claxton and Murrell 1987; Melear 1990).

The **Learning style profile** was produced by the National Association of Secondary School Principals (NASSP) (Keefe, Monk, Letteri, Languis, Dunn 1989). Learning style was defined by Keefe and Monk (1990, 1) as

the composite of characteristic cognitive, affective, and psychological factors that serve as relatively stable indicators of how a learner perceives, interacts with, and responds to the learning environment. It is demonstrated in that pattern of behaviour and performance by which an individual approaches educational experiences. Its basis lies in the structure of neural organisation and personality which both moulds and is moulded by human development and the learning experiences of home, school, and society.

The Learning Style Profile (LSP) contains 126 items measuring 24 subscales and provides easily utilised information on the cognitive elements of style, as well as measures of perceptual, affective, and environmental styles (Jenkins, Letteri and Rosenlund 1990, 1). Cognitive controls (skills) are internal to the human information processing system and are instrumental in all learning. If a student has good cognitive skills, he or she is ready to learn efficiently and successfully. If a student lacks these skills, frustration and failure are the likely outcomes.

The LSP subscales relevant to the research question are the cognitive skills subscales (analytic, spatial, discrimination, categorisation, sequential processing, simultaneous processing and memory) and the persistence orientation subscale. Leino, Leino, and Lindstedt (1989, 1) have shown that the cognitive style profile subscale of the LSP correlates highly with school achievement, but only a few subscales of other sectors of the LSP correlate with school achievement.

Analytic skill is measured by the ability to identify simple figures hidden in a complex field, the use of the critical element of a problem in a different way

(Keefe and Monk 1990, 5). Analysis is a skill that refers to an understanding of the whole task based upon the breaking of the problem into its salient parts. Jenkins, Letteri and Rosenlund (1990, 77) suggest that most school-related achievement requires the skill of analysis.

The field-independent versus field-dependent continuum is a frequently used indicator of analytical skill. Field-independent learners tend to impose structure on a field if a logical pattern does not exist, whereas field-dependent learners accept the field the way it is (Witkin, Moore, Goodenough, and Cox 1977). Field independent learners appear to have greater skill in 'cognitive restructuring'; that is, in understanding problems and reformulating problems into structured ideas (Foreman 1988, 5). The field dependent/independent dimension was first identified by the Rod and Frame Test (RFT). The image of a rod and a frame was placed in a dark room with the frame tilted. Subjects were then asked to move the rod so that it was placed in a vertical position. Field-independent learners placed the rod in the vertical position independent of the frame, a perception independent of the environment, while field-dependent learners aligned the rod with the frame, a perception dependent upon the environment (Witkin et al., 1977). Field dependence/independence is more commonly measured by the Embedded Figures test (EFT) or Group Embedded Figures Test (GEFT) which measures subject performance on a series of problems in which the subject must find a simple figure in the context of a complex set of figures (Foreman 1988, 5; Bishop-Clark 1992, 3).

Jenkins, Letteri and Rosenlund (1990, 1) submit a connection between field independent behaviour and the cognitive skill of analysis. This is supported by Pemberton's (1952) statement that field independent subjects were interested in 'analytic' endeavours. The term 'analytical' was used by Piaget (1954) to describe the perceptual behaviour of older children and adults as opposed to the relatively field dependent perceptual behaviour of young children (Witkin 1977). Bishop-Clark (1992) reports a meta-analysis of correlational studies between field-independence and programming achievement (Table 2). The studies have emanated from theoretical suggestions that people who are able to impose structure on seemingly unstructured situations (field-independent) will have greater success in computer programming activities.

Study (year)	no.	r-inde	exgrade	language	instrument
Bradley(85) Cavaini(89)	26 30	.63 .29	elementary university	logo cobol	Hidden Patns T GEFT
Cheney(80)	35	.82	university	basic	Barkin instru.
Foreman(88)	46	.39	university	basic	GEFT
Hassell-a(82) Hassell-b	29 19	.07 .49	university	fortran	EFT
Stevens(83)	73	.35	university	basic	GEFT
Testa(73)	34	.80	university	cobol	EFT
Webb(84)	35	.45	junior high	logo	Hidden Patns T
Werth(86) Wilson-a(90) Wilson-b	58 58 58	.32 .47 .33	university elementary elementary	pascal logo logo	GEFT GEFT GEFT

Table 2.Meta-analysis of correlational studies between programming
achievement and field-independence (Bishop-Clark 1992).

Bishop-Clark (1992) concluded that although there is not an established causal link between field-independence and higher grades, the studies clearly revealed that field independence is positively related to achievement in computer programming classes.

Spatial skill refers to the identification of geometric shapes, the rotation of objects in the imagination, and to the recognition and construction of objects in mental space (Keefe and Monk 1990, 5). Students with strong spatial skills are able to see distinguishing characteristics that other students miss and are able to create spatial models to represent concepts (Jenkins, Letteri, and Rosenlund 1990, 9). The same authors suggest that strong spatial skills are likely to support success in subjects such as mathematics, geography, chemistry and physics.

Sex differences in spatial skill have been found in a large number of studies (Harris 1981). Males would appear to be superior in spatial skill tasks such as rotational tasks (Metzler and Shepard 1974), in the folding of a two-dimensional geometric pattern into a three-dimensional figure (Bennett, Seashore, and Wesman 1959), and in a sense of direction or way finding (Harris 1981).

Sutton-Smith (1977) and McGuiness (1976) consider that, as children grow older, the sex differences in spatial skill recorded earlier in life are reduced even if not entirely eliminated. It is unknown whether spatial skills are amenable to change (Jenkins, Letteri and Rosenlund (1990, 9).

Discrimination skill refers to the visualisation of the important elements of a task, the focusing of attention on required detail, and to the avoidance of distractions (Keefe and Monk 1990, 5). Students with strong discrimination skills are able to focus on the important components of a task and filter the relevant details from the irrelevant details (Jenkins, Letteri and Rosenlund 1990, 9).

The original research on discrimination referred to the skill as 'focusing' (Schlesinger 1954). Schlesinger (1954) saw focusing, an initial description of the discrimination skill, as involving a tendency to narrow awareness and a tendency to separate emotion from the idea. Gardner, Holzman, Klein, Linton, and Spence (1959) described focusers as individuals who deploy attention and scan many aspects of experience rather than only a few. Narrowness of view was described by Piaget, Vinh-Bang and Mantalon (1958) as the cause of both underestimation and overestimation of the size of certain objects (Jenkins, Letteri and Rosenlund 1990, 17).

Categorisation skill refers to the use of reasonable versus vague criteria for classifying information, to the formation of accurate, complete, and organised categories of information (Keefe and Monk 1990, 5). Gardner (1953) together

with Bruner, Goodnow and Austin (1956) have demonstrated that individuals may be consistently characterised by category width differences in a variety of adaptive tasks. Pettigrew (1958) concluded that a person's tendency toward broad, medium, or narrow categorising remained consistent over a wide range of tasks.

Students with narrow categorisation skills tend to use precise criteria in identifying new information. Letteri (1985, cited in Jenkins, Letteri and Rosenlund 1990, 28) found that narrow categorisers do better in school because they are able to identify new information with enough precision to place it accurately into the network of categories of long-term memory.

Sequential processing skill refers to the processing of information sequentially and verbally, and to readily deriving meaning from information presented in a step-by-step, linear fashion (Keefe and Monk 1990, 5). *Simultaneous processing skill* refers to the grasping of visuo-spatial relationships; to sensing an overall pattern from the relationships among component parts (Keefe and Monk 1990, 5).

The Russian psychologist Alexander Luria (1973) laid the foundation of sequential and simultaneous processing. Luria advanced the view that the brain is composed of three functional units or blocks, each with a structural component of three levels. Each block has a specific function to perform and is related to the other blocks. Block two is located in the occipital, parietal, and temporal lobes of the right and left hemisphere. It is in this block that information is processed in two qualitatively different ways, simultaneous and successive. Strength in both modes of processing was deemed crucial to successful performance, and each was required in different degrees to perform all tasks (Jenkins, Letteri and Rosenlund 1990, 35).

Strength in sequential processing, and in simultaneous processing is related to high achievement in school (Das 1973; Krywaniuk 1974; Sprecht 1976; Kirby and Das 1977; Jenkins, Letteri and Rosenlund 1990, 35). Das, Manos, and Kanungo (1975) noted that lower socioeconomic children demonstrated a preference for the successive mode. A preference that suggests that sequential and simultaneous processing skills are amenable to change.

Das, Kirby, and Jarman (1979) constructed a set of tests to measure simultaneous and successive processing and concluded that the two kinds of processing are distinct cognitive styles, related to such constructs as field dependence-independence, reflectiveness-impulsivity, and conceptual level (Keefe and Languis 1989, 4). Keefe and Languis (1989, 4) make the point that simultaneous and successive processes refer to the ways individuals deal with tasks mentally and not to the nature of the tasks themselves.

Memory skill refers to the retention of distinct versus vague images in repeated tasks and to the detection and remembering of subtle changes in information (Keefe and Monk 1990, 5). Students with strong memory skills are able to recall accurate information when required to do so. Success in school is closely related to skill in remembering information accurately (Jenkins, Letteri and Rosenlund 1990, 47).

Memory skill research has used descriptors of sharpening and levelling when referring to strong and weak memory (Jenkins, Letteri and Rosenlund 1990, 47). Sharpeners are better able to recover past experiences than levellers because levellers minimise differences and tend to assimilate new information into that previously experienced. (Gardner, Jackson, and Messick 1960; Holzman and Gardner 1960). It is likely that memory skill is amenable to change because levelling has been observed to be more characteristic of younger children and sharpening more characteristic of older ones (Santostefano 1985; Jenkins, Letteri and Rosenlund 1990, 47).

Motivation

The proposed research intends to investigate whether the use of object-oriented programming and its associated inherited user interface will add value to the learning and thus make the course more enjoyable and intrinsically rewarding for most students. Recent theory and research on motivation based upon a model of expectancy times value (Feather 1982; Good and Brophy 1991) has been mentioned previously.

The Academic Motivation Scale (Vallerand 1992), which measures intrinsic and extrinsic motivation in education and the Miller Motivational Scale (Miller 1987) show some promise as feasible measuring instruments. The Academic Motivation Scale is a seven-factor measure of motivation toward education which has reported satisfactory validation and reliability. The Miller Motivational Scale appears to be less appropriate to the research question in that it places more emphasis upon measuring personality associated motives.

Other reports of intrinsic value are inferred by performance and course selection rather than an attempt to directly measure motivational value (Seymour, Sullivan, Story, and Mosley 1986; Armour, White, and Boehm 1987). While the most effective elements of motivation are unknown, research by Chiu Lian-Hwang (1967) has outlined five factors which may be used to derive a motivational profile:

Does the student have a positive orientation toward learning? Does the student show persistence and a high level of aspiration? Does the student have positive feelings toward personal academic self-concept and past performance?

Does the student manifest a need for academic recognition from teachers and peers?

Does the student fear failure and try to avoid it to a reasonable degree? (Under achievers may be too anxious or too bored to learn well.)

Is the student curious, both about concepts and about things?

Does the student work when demanded by the teacher, or by parents, or even by friends? Is the student responsive to authority and peer influence?

As earlier outlined in an analysis of learning style, the affective component of learning style includes among other elements motivation, expectations and the degree of interest in the subject matter to be learned (Claxton and Ralston 1978; Smith 1982). The measurement of motivation would thus appear to be subsumed within the determination of learning style.

Persistence orientation, an affective subscale of the Learning Style Profile, refers to a willingness to work at a task until completion. Persistence is influenced by achievement motivation, sometimes referred to as intrinsic motivation, and is stimulated by any task that is worthwhile in itself (Keefe 1989, 5). Singer, Korienek, Jarvis, McColskey, and Candeletti (1981) suggest that persistence can be improved with moderate and positive feedback to the student, particularly feedback based on the individual's goals. Factors such as achievement motivation - the desire to excel, the need to identify with other successful students, and adult example, all affect persistence (Clarke 1972). Persistence can be aroused either by the expectation of success or the fear of failure. Persistence is related to risk taking, using concrete and immediate feedback to modify personal goals and behaviour, assuming personal responsibility, and sizing up the environment for its limits and possibilities (Alschuler, Tabor, and McIntyre 1971). There appears to be considerable evidence suggesting that motivation and achievement increase when the instruction is matched to the learner's preferred learning modes (Dunn 1980; Jenkins 1982; Pizzo 1982; Curry 1983; Dunn 1984; Claxton and Murrell 1987; Melear 1990).

Attitude

The anticipated intrinsic motivational value of students being able to produce sophisticated user interfaces for their programs suggests the possible use of an attitude towards programming scale.

It is asserted by Oppenheim (1992, 174) that a majority of researchers define an attitude as 'a state of readiness, tendency to respond in a certain manner when confronted with certain stimuli.' Support for Oppenheim's assertion is given by Aiken's (1980, 2) definition of attitudes as 'learned predispositions to respond positively or negatively to certain objects, situations, concepts, or persons'. Similar definitions have been offered by Fishbein and Ajzen (1975). This general definition of attitude is further qualified as

Attitudes are reinforced by beliefs (the cognitive component) and often attract strong feelings (the emotional component) which may lead to particular behavioural intents (the action tendency component) (Oppenheim 1992, 175).

The measure of attitude to(wards) programming, using easy-to-administer Likertscale instruments, is dependent upon a model of the relationship between attitude and computer usage and achievement. The model assumes that attitudes precede and predict behaviour. Fostering positive computer attitudes would be expected to increase computer usage and achievement (Richards, Johnson, and Johnson, 1986; Bear, Richards, and Lancaster, 1987; Todman and File, 1990). The link between attitude and behaviour hasn't been established (Mishler, 1984; Shrigley, 1990) and any measure of attitude, assumed to reflect behaviour in a computing environment, must therefore be approached with caution. Oppenheim (1992, 175) also makes the point that the model is primitive in that it is assumed that attitudes are measured as a linear continuum from positive through neutral to negative. The model, however, is convenient for making measurements.

Sutton (1991) lists seven ways in which measurements of attitude towards computers have been defined. The various definitions of attitudes towards computers have included male domain, general, interest, liking, utility, confidence, and anxiety. Kay (1993) reports the use of fifteen different constructs used in computer attitude studies. The various constructs were enjoyment, anxiety, efficacy, gender-typing, policy concerns, educational support and benefits, computer use, computer-aided instruction, potency of computers, helpfulness, awesomeness, negativity, and science fiction.

There is a suggestion that definitions of attitude towards computers involving confidence, liking and interest show less gender differences than male domain, anxiety, utility and general attitudes (Morse and Daiute 1992, 4). Kay (1993, 372) strongly counsels that attitude measures be developed from a theoretical base to provide a unified foundation and to allow for easier interpretation.

The difficulty in establishing a measure of attitude towards programming is in isolating it from a myriad of other attitudes related to computers in general with which it is associated and intertwined. The drawing out of constructs to be used in an attitude scale needs to be cognisant of the theoretical attitudinal model (affective, cognitive and behavioural components), the context (programming) and purpose of the attitude instrument (intrinsic motivational value).

An attitude to(wards) programming scale was developed as a part of previous coursework. Three constructs - *liking of programming, programming difficulty*,

and *programming usefulness* - derived from the context of this study and the theoretical attitudinal model of affective, cognitive and behavioural components were incorporated in a pilot fifty-three item Likert-type scale of attitude to(wards) programming.

Programming instruction

Pea, Kurland and Midian (1984, 13) set two questions about programming instruction: how much direct instruction and what type of instruction should be offered?

The effects of varying proportions of direct instruction and computer access required to support programming instruction has not been thoroughly researched. It would appear important that computer access, ie practical experience, support classroom instruction. It is also conjectured that unlimited access, associated with small amounts of direct instruction, may be associated with little program planning and consequently poor techniques and poor conceptual understanding (Mayer 1979; Pea, Kurland and Midian 1983; Dalby, Tournaire, and Linn 1986). Ross and McCormick (1989, 8) suggest that imposing reasonable limitations on computer access may encourage students to give greater attention to program design and mental execution of code away from the computer.

Deimel and Moffat (1982) and Pea (1986) proposed that an introductory programming course should not concentrate on the writing of programs but on the reading, modification, and amplification of non-trivial, well-designed working programs. van Merrienboer (1988, 184) believes that reading templates demonstrates direct application, and that modification of templates requires a thorough understanding of structures combining several language features. The use of templates promises an approach to programming instruction that can result in better student achievement. Johnson and Anderson (1985, 697) support the template approach to the teaching of programming in that embedding key concepts in complete modules, which accomplish a particular task, is more appropriate than the piecemeal statement by statement assembly approach. The template approach supports additional practice in reading and generalising, the demonstration of interesting and powerful programs, with the added benefit of promoting a more structured approach to the design of a program (Johnson and Anderson 1985, 697).

The use of templates within a guided instructional environment may also have implications in assisting the explicit teaching of problem-solving strategies. Problem-solving in programming involves the design of problem solutions and the development of template repertoires (Linn, Sloane and Clancy 1987, 470; Linn, Sloane and Clancy 1987, 474). Medium to low ability students have been shown to achieve greater problem-solving success within an explicit instructional environment than in unguided discovery classes (Doyle 1983, Eylon and Helfman 1985, Dalbey and Linn 1985).

Further support for the use of templates for students to read, analyse and modify is provided by Sleeman, Putman, Baxter, and Kuspa (1984) and Sheil (1981). The detail of the language's syntax and semantics would then be introduced as secondary issues.