Technicians and their Learning Styles Preferences and Cognitive Processes of Formal Inferences

Luis Reynoso, Eduardo Grosclaude, Laura Sánchez
Faculty of Computer Science
University of Comahue, Argentina
Buenos Aires 1400 (8300) Neuquén
{Luis.Reynoso, Eduardo.Grosclaude, Laura.Sanchez}@fai.uncoma.edu.ar

Mabel Álvarez
Patagonia San Juan Bosco University, Argentina
Belgrano y Rawson (9100) Trelew, Chubut
mablop@speedy.com.ar

Abstract

During the last seven years several Argentinian national universities have offered (as part of their academic studies and programs) different undergraduate degrees under the denomination of technical degrees. The personal characteristics of their students are radically different from the traditional academic offer. Students learn by practicing and they comprehend information best by actively doing something with the information. We ran an experiment with students of a technical degree of web development in a course of Programming Introduction using Python. Preliminary findings revealed that their learning styles are mainly active and visual, and learners who are more verbal or have stronger concrete experience obtained higher scores in their tests. They perceive that inductive tasks are easier than deductive and abductive tasks. We also found that those subjects who are more efficient in solving formal inference tasks obtained higher qualifications in their exams. The findings can be useful not only for didactic transposition in teaching courses which take into account the balance of the students' preference but also to develop new instructional methods and software which focus on the cognitive preferences and cognitive process of technicians.

Keywords: Learning Style Models. Formal Inferences. Cognitive Process. Information Acquisition.

Proc. 12th IEEE Int. Conf. on Cognitive Informatics & Cognitive Computing (ICCI*CC'13)
D.F. Hsu, Y. Wang, A.R. Rao, D. Zhang, W. Kinsner, W. Pedrycz, R.C. Berwick & L.A. Zadeh (Eds.)
978-1-4799-0783-0/13/$31.00 ©2013 IEEE

1. INTRODUCTION

During the last seven years several Argentinian national universities have offered as part of their academic programs undergraduate degrees under different denominations of technical bachelor's degrees. The degrees do not provide a priori a joint curriculum with the graduate academic offer. The technical undergraduate degrees can be obtained after few years of studies during which technical skills and knowledge in a specific topic or area is applied. These degrees enable faster job prospects, and they allow to compete against a growing demand for short degrees offered until a few years ago only by non-university colleges. The Secretary of University Politics of the Ministry of Education fostered the creation of these technical degrees. The minimal timespan of these undergraduate degrees is 1600 hours spread over two and a half years [1].

The formation of technicians is considered a prioritary strategy for the economic and productive development of the country (Argentina). Students of these technical degrees can also apply to scholarships provided by the Ministry of Education, due to the fact that the formation of technicians is part of an academic policy to cater for a deficit of formed and specialized technical staff in enterprises and state organizations.

During the last five years different technical degrees have been part of the academic offer of the Computer Science Faculty of National University of Comahue, Argentina: technical degree on web development, technical degree on programming and database management, technical degree on system administration and free

software. Students of technical undergraduate bachelor's degrees are different from those of graduate degrees. The way of teaching is also different, the didactic transposition needed in technical degrees is higher than that applied for graduate degrees; scientific knowledge should be presented through practice and case studies. The audience demanding these academic offer is also different. Their preferences and learning style is based on practice of real cases, and they learn better through examples rather than theory.

Felder reported the problems that ocurr when the learning styles of students and professors do not match. He also reported the benefits when instructional methods are balanced according to their preferences.

This motivated us to run an experiment to observe and describe a group of technicians studying web development, to analyze their learning styles and their influence in final scores, and to study their perception about the complexity of the three more important cognitive processes of inferences. Our hypothesis is that the subjects will rate inductive cognitive process as easier than other cognitive processes (such as deduction and abduction) because technicians elaborate theory up from cases dealt in practice. We are also interested in finding out whether a correlation exists between subjects who are more efficient in applying cognitive process and subjects who score higher in the final exams.

We are interested in studying the learning style of individuals to conclude about the group characteristics. Their preferences, taken together, define who they are and how they act. Then the didactic transposition can be adapted accordingly.

The paper is organised as follows. Section 2 presents the basic learning style tests we applied in this study. Section 3 provides an overview of the cognitive processes of formal inferences. Section 4 describes the previous work. Section 5 describes the experiment undertaken, its context and design, results and interpretations. Section 6 goes on to describe conclusions and future work. The appendix contains extra information concerning the experiment design.

2. LEARNING STYLES

2.1 KOLB LEARNING STYLE

According to Kolb the learning style preference itself is actually the product of two categories: *processing* and *perception*. Processing and perception reflect the major directions of cognitive development derived from the work of Piaget [2]. Each Kolb category is based on two pairs of variables, or two separate 'choices' that we make, which Kolb presented as lines of axis (see Figure 1), each with 'conflicting' modes at either end:

- Processing: Active Experimentation - AE (doing) vs. Reflective Observation - RO (watching)
- Perception: Concrete Experience - CE (feeling) vs. Abstract Conceptualization - AC (thinking)

The combination of these two categories sets out four distinct learning styles (or preferences) in the Kolb's learning theory:

- Diverging (CE/RO): Divergents tend toward a concrete experience and reflective observation.
- Assimilating (AC/RO) Assimilators are characterized by abstract conceptualization and reflective observation.
- Converging (AC/AE) Convergers are characterized by abstract conceptualization and active experimentation
- Accommodating (CE/AE) Accommodators use concrete experience and active experimentation.

2.2 FELDER LEARNING STYLE

The model was originally formulated by Dr. Felder in collaboration with Dr. Linda K. Silverman, an educational psychologist, to be used by college instructors and students in engineering and the sciences, although it has subsequently been applied in a broad range of disciplines. It basically uses the pair of Kolb's categories and adds two more categories: *input* and *understanding*.

- Processing: Active or reflective (ACT/REF). Active learners tend to retain and comprehend information best by doing something active with the information- discussing it or testing it in some way or explaining it to others, whereas reflective learners learn best by thinking things out on their own. Active learners prefer to work in groups whilst reflective ones prefer to work alone.
- Perception: Sensing or intuitive (SEN/INT). Sensing learners tend to be more practical and careful than intuitive learners, and they prefer learning facts and solving problems with well-established methods. Conversely, intuitive learners tend to be more innovative than sensing learners and better at grasping new concepts
- Input: Visual or verbal (VIS/VER). Visual learners retain more from pictures, diagrams, flow charts, time lines, films, and demonstrations; in

contrast, verbal learners learn better out of written word or spoken explanations.

- Understanding: Sequential or global (SEQ/GLO). Sequential learners tend to learn in linear, logical steps; on the other hand, global learners prefer to learn in large jumps, absorbing random pieces of material, and they suddenly 'get it'.

3. COGNITIVE PROCESS OF FORMAL INFERENCES

Several studies have investigated the formal inferences and their utilization in the design process. From a logical perspective, reasoning in knowledge engineering follows certain patterns: induction, deduction, abduction, abstraction. These concepts provide a guiding structure for all abstractions or application of knowledge [3].

Induction shows that something actually is, deduction proves that something must be, abduction suggests that something may be.

We will briefly describe these cognitive processes:

- Induction: Induction is a process of reasoning, concluding from one or more specific cases to a general principle [3]. It is concerned with finding unifying patterns. Given certain facts, it is assumed that some general principles unite these facts. The objective of the designer is to find the rules by which the facts are connected. His/her main concern is to explain the rules by linking them with other general facts.

- Deduction: Deduction is also a method of reasoning by inference from premises which is defined as inference by reasoning from a general rule to a particular solution. In this method, the designer sets up a system consisting of general rules, and deduces a particular solution from it [4]. Concluding from a general principle to a specific case [3].

- Abduction: Peirce describes abduction as the spontaneous conjecture of instinctive reason [5]. Abduction consists of studying the facts and then devising a theory to explain them.

  Abduction is a process of productive reasoning. It is the inference of a case from a rule and result, since it reflects the designer's presumption that a certain phenomenon might exist to account for his/her observations. It is where the designer can find that, in certain respect, two objects have strong resemblance, and infer that they resemble one another strongly in other respects [4]. Inventing a new general principle by deriving a hypothesis from a special case [3].

  Abductive reasoning plays the role of a generator of new ideas or hypotheses. It has been shown to be specially useful as a mechanism for the detection and diagnosis of inconsistencies in different fields of software engineering [6].

- Abstraction is a process to elicit a subset of objects that share a common property from a given set of objects and to use the property to identify and distinguish the subset from the whole in order to facilitate reasoning [7].

To understand the importance of applying cognitive process, first we propose to understand the difference between the fact of describing and explaining. For that purpose we refer to Schuster [8]: When we describe we indicate recognizable hallmarks of things. The fact of describing implies remaining at the same propositional level than what we are describing, there is no change of plane. Instead, the explanation does imply a change of propositional level. To explain is to subsume [8], incorporate a fact under a general statement.

Every time we apply a cognitive process of formal inference we are dealing with at least two propositional levels, as we do when we explain something. Cases or facts in one propositional level, and general statements or rules in another level. The transfer from one level to another rests in part on perceptual decontextualization of structure from content, use of formal logic [9], finding patterns, and avoidance of contradiction [10].

Decontextualizing [11] is the handling of information in a way that either disconnects other information or backgrounds it [12], it is produced when the meaning of the signs is becoming less dependent on the spatial and temporal context in which they are used. For example, when we obtain (to induce) a general rule for a set of observed facts we abstract/produce a common structure behind the observed cases, so changing of propositional levels is applied.

These informal inferences are constantly applied in real life and computer science discipline. When programming, making variables to stand for objects in the problem, and making operations to stand for relationships among objects in the problem, are clearly usages of formal inference. These inferences are also present in the transformation of elicitation process into system specification, between different levels of specification of models, between correspondences of executions and programs, between models and metamodels of a system, between object and classes, etc.

4. PREVIOUS WORK

To provide a description of the previous work related with the aim of this study we must focus on two different aspects: the relationships between learning styles and learning performance in computer science courses, and the relationships between cognitive process and student's performance.

Regarding the first studies, it is possible to find several researches under the premise that taking the students' learning styles into account, helps to enhance teaching effectiveness which in turn, improves the student performance. Many of these studies had shared a common conclusion: they found that *reflective* and *verbal* learners outperformed active and visual learners (Allert [13], Chamillard and Karolick [14], Thomas et al. [15], Zualkernan et al. [16]). For more details we refer the reader to [17].

In relation to the second aspect, cognitive informatics is a cutting-edge and profound interdisciplinary research area that tackles studies of natural intelligence and internal information processing mechanisms of the brain, as well as the process involved in perception and cognition. Wang describes in [7] the cognitive process of formal inferences (induction, deduction, abduction and analogy) and he specifies a set of mathematical models of formal inferences methodologies. A formal description of four forms of cognitive processes of inferences are also presented in [7].

5. AN EXPERIMENT

In the following we will describe the context of the experiment:

5.1 Operation

The operational phase of the experiment is divided into three steps: preparation, execution and data validation.

- Preparation. We have selected as experimental subjects a group of students who had taken a semester class on Programming Introduction offered by the Computer Science Faculty, at Comahue University. Hence, the experiment was run off-line. In this course the students had learned the Python programming language. The students were asked to participate in the course, 24 subjects agreed to take part, so they were volunteers. They were motivated to do some practical exercises but it was not mentioned that these exercises were constituted an experiment. The subject were not aware of what aspects we intended to study. Neither were they aware of the actual hypothesis stated. In the lecture before the experiment was carried out, the subjects were asked to do two learning styles' test (Kolb's and Felder's Tests).

  We prepared the material handed to the subjects, consisting of:

  – Hand-on programming tasks: Students were asked to write a complete Python program according to a set of software requirements. A sample of this kind of task is shown in appendix B.

  – Written-tasks: Each subject was asked to complete six tasks. The set of the six tasks consists of three pair of tasks, where each pair corresponds to different cognitive process (induction, deduction or abduction). Appendix A shows an example of each one. We will explain them in the next paragraph.

  Before running the experiment we performed a pilot experiment. We asked a researcher who has experience in Python to carry out the experimental tasks. All the modifications she suggested were considered.

  Each written assignment consisted of some Python code and included a test with two different tasks:

  – A multiple choice test related to a task which involved a cognitive process (deduction, induction or abduction). The complexity of the Python code was similar. They also had to write down the time they started to do the task and when they finished.

  – Rating Tasks: The subjects had to rate each task, using a scale consisting of four linguistic labels (Very difficult to comprehend, a bit difficult to comprehend, A bit easy to comprehend, and very easy to comprehend.)

  Hand-on programming tasks were solved by the subjects during a week. Written tasks were solved during an experimental session. In the session the subjects were able to use a clock rendered with a multimedia projector. The subjects were given all the six tests described in the previous paragraph. We explained to them how to carry out the test, asking for carrying out the test alone, and using unlimited time to solve it. There was an instructor who supervised the experiment, if the subjects were in doubt, he could be asked. We collected all

the data, including subject' rating obtained from the responses of the experiment.

We will describe the written-tasks of the experiments:

- Induction Tasks (IND-Tasks): Salama describes that induction may have three underlying steps: observing a large number of facts, formulating theories to explain these facts, and testing the theories by experimentation [4]. The IND-Tasks consists of three program executions of a Python program. We bring as IND-tasks three programs executions and three different programs, the subject must observe the program executions and realize to which of the three programs they correspond.
- Deductive Tasks (DED-Tasks): We tried to design DED-Tasks where the subject reasoning from a general rule to a particular solution. We included a piece of python code (this code can be thought as a general solution for a class of problems) and three different executions. Only one of the three executions corresponds to the piece of code provided. The subject must deduce which of the three executions is the right one.
- Abduction tasks (ABD-Tasks): They should consist of studying different facts and devising a theory to explain them. We provided them with three incomplete executions of a program along with three different pieces of Python code. The execution of the program shows a series of numbers printed by the program, however, the list of numbers printed was not completed. We did not show the whole execution of the program. The subject should explain which of the three incomplete execution sequences corresponds to which program. Only one execution may correspond to only one program. We said *may* because the execution was shown partially and the subject should only abduct to which code it may correspond. The source program only could have been positively identified if the execution sequence had been wholly disclosed. Nevertheless, the remaining execution sequences and programs might not belong to each other for different reason. So, only one correspondence may be established and justified.

- Data Validation: Analyzing the learning style test, we can corroborate that almost all the subjects share the same Felder's learning style subcategory of processing and input: they are Active and Visual. So, empirical findings about reflexive and verbal (described in section 4) are not useful in our context.

  Not all the tests were correctly answered. In average, less than 30 % of tasks were incorrect. We think that this information can not be separated and should be analyzed. Finally, we had 96 data sets to be analyzed. We score the test according to its correctness in a scale of 1-10.

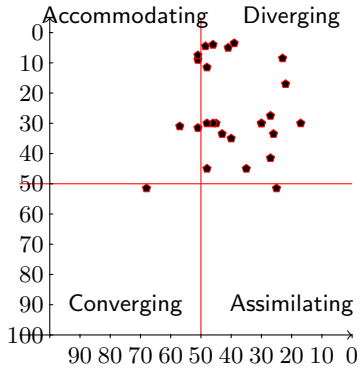

Figure 1. Kolb axis

5.2 ANALYSIS AND INTERPRETATION

5.2.1 KOLB LEARNING STYLE

Once we had analyzed the Kolb test we found that 75% of the subjects are *accommodating* (see Figure 1). Accomodating people tend to perceive the specific information and process it actively. They integrate experience and application. They learn by trial and error. They believe in self-discovery. They are flexible and they enjoy the change.

5.2.2 FELDER LEARNING STYLE

From Felder's Test we obtained that 75% of the subjects are *active* and 25% of the subjects are $reflective$, 83% of the subjects are *visual* being 17% *verbal*. Regarding the percentages of *Intuitive*/*Sensitive* are 37/63, and *sequential*/*global* percentages are 50/50.

5.2.3 KOLB AND FELDER LEARNING STYLE

- The left and right quadrants of Figure 1 correspond directly to Felder's Active-Reflective scale.

  Test. Spearman correlation between Kolb's EA/OR scale and Felder's Active-Reflective scale: $H_{0,1}$: There is no significant correlation between Kolb's EA/OR scale and to Felder's Active-Reflective scale. $H_{1,1}$ = not $H_{0,1}$.

  To test the hypothesis a correlation analysis was performed using Spearman's correlation coefficient as this statistics measure the rank-order association between two scales or ordinal variables. To determine if the rho coefficient is statistically significant (eg. Reject the Null hypothesis) we compare the magnitude of rho versus the critical values of spearman's rank correlation coefficient with degree of freedom = 22, at the 0,05 level of significance (Table 1 shows the results). We used a level of significance alpha = 0.05 which means the level of confidence is 95%. A rho correlation coefficient of 0.428 is required for statistical significance. Thus, the observed rho of 0.52 indicates that there is a relationship between Kolb's EA/OR scale and to Felder's Active-Reflective scale.

- The top and bottom quadrants of Kolb's scale correspond directly to Felder's Sensing-intuitive scale.

  $H_{0,2}$: There is no significant correlation between Kolb's CA/EC scale and to Felder's Sensitive/Intuitive scale. $H_{1,2}$ = not $H_{0,2}$.

  To test the hypothesis a correlation analysis was performed using Spearman's correlation. A rho correlation coefficient of 0.361 is required for statistical significance at a 0,10 level of significance. (We used a level of significance alpha = 0.1 which means the level of confidence is 90%, see Table 1. Thus, the observed rho of 0,4096 indicates that there is a relationship between Kolb's EA/OR scale and to Felder's Active-Reflective scale.

5.2.4 ANALYSIS OF THE EXPERIMENTAL TASKS

- The average time the subjects spent in IND-, ABD, and DED-tasks are 05:41, 05:00, 04:53 minutes respectively. Although their times are fairly similar, the order is similar to the rating of the task's complexity, which is explained in the following paragraph.

Table 1. Correlation Tests

Test	Rho Correlation
Kolb's EA/OR & Felder's ACT/REF	0.5195
Kolb's CA/EC & Felder's SEN/INT	0.4096

- DED-tasks were rated as more difficult than IND-tasks and ABD-tasks; and between the last two, ABD-tasks were perceived as more difficult than IND-tasks. The subjective complexity perceived for IND-, ABD- and DED-tasks are 2.32, 2.24 and 2.17 respectively. We think that this is the reason why, due to the fact that most of them are actives, and for active people solving inductive tasks is easier, as they can detect the general rules reading different examples. Active people learn from practice, *by doing or applying things.*

- As we previously mentioned most of the subjects were active and visual, nevertheless after applying the Felder Test the score of ACT/REF, INT/SEN, VIS/VER and SEC/GLO in a range between -11 and 11 (where a positive number indicates that a person belongs to the first dimension, and a negative one to the second dimension) was used to find correlations with two tests scores (the final score in the course for the subject, and the score of the hand-on programming tasks). The Spearman correlation is shown in the first three rows of Table 2. A similar correlation was done with the Kolb test (see the last two rows and first three columns of Table 2).

- As we previously mentioned the subjects performed a hand-on programming activity. This task consisted of a practical activity of developing a Python program during one week. The subjects were invited to develop the program in groups (of two subjects) or alone. Different requirements were specified for each group in order to avoid copies of the program. This election (work with a partner or alone) was also correlated with ACT/REF dimensions, due to the fact that ACT learners tend to like group work more than REF learners, who prefer working alone. We found a spearman correlation of 0.741, so this assertion is true for our group of subjects.

- From Table 2 we can observe that only the dimensions of Felder's VIS/VER and Kolb's CA-EC correlate between these tests, and this correlation is negative. That means those subjects which are more VERBAL obtained higher test scores. Similarly those subjects who had higher values of EC (concrete experience) obtained better results in both tests.

- We also performed a correlation study between the two kinds of tests with the cognitive process activity. See last two rows and columns of Table 2. The six tasks each subject performed were examined and rated in a scale of 1-10. We found a significant correlation between them.

Table 2. Spearman Correlations

	ACT/ REF	INT/ SEN	VIS/ VER	SEC/ GLO
written test	0.388	0.046	**-0.826**	0.159
hand-on progr. task	0.390	-0.097	**-0.897**	0.108
	EA/ OR	CA/ OR	IND- DED- ABD- Tasks	
final test	0.343	**-0.690**	**0.809**	
hand-on progr. task	0.281	**-0.770**	**0.800**	

6. CONCLUSIONS

Ten years ago, learning style theories had been criticized due to the fact that there was little evidence for the efficacy of most learning style models and for the dubious theoretical underpinnings [18]. However during recent years, many empirical studies have shown the importance of taking students' learning styles into account to enhance teaching effectiveness [13], [14], [15], [16]. The aim of this study (an experiment) is to observe correlations between learning styles and students performance in a particular group of college freshmen. We also study the correlations between cognitive processes and students performance. The students were enrolled in an academic undergraduate program of technical studies on Web Development.

We asked a group of twenty four students to fill out two Learning Styles questionnaires: Kolb and Felder tests. According to the former learning styles most of the technician students are *diverging* whilst the latter classified the students as *active* and *visual*. We found this normal because technician students tend to be more practical, and those who study web development are more visual. The higher their score in Kolb' concrete experience, the higher their performance in tests. We also found that visual/verbal Felder' dimension was negatively correlated with students' performances in tests, meaning that those college freshmen who are more verbal outperformed visual ones.

We also found correlations between the two tests (Kolb and Felder) where their dimensions examine the same learning aspects (Felder's ACT/REF and Kolb's EA/OR; Kolb's CA/EC and Felder SEN/INT). In addition we tested that ACT/REF learners are correlated with student preferences of working alone or in group.

Moreover, unlike most studies that only examined correlations between learning styles and student performance this study has attempted to investigate whether cognitive processes of *induction*, *abduction* and *deduction* might be considered good indicators of learning outcomes. Correlation was found between these two aspects. Furthermore students' perception of tasks applying different cognitive processes indicates that technicians consider inductive tasks as easier than abduction and deduction tasks. Inductive tasks could be easier to solve for a technician who elaborates theories from cases or practice.

All the results can be considered as preliminary and replication of the experiment should be run to extend its external validity. We plan to replicate the experiment in a course of the technical bachelor's degree of Software Administration and Free Software.

ACKNOWLEDGMENTS

This research is part of the 048/12 'Hacia el Fortalecimiento de la Sociedad en el Uso y Aplicación Geoespacial y las TICS' of Patagonia San Juan Bosco University and the 'Modelos y Tecnologías para Gobierno Electrónico' of Comahue University (Argentina).

References

[1] S. de Politicas Universitarias, "Criterios y Procedimientos. Evaluación o Modificación de una Carrera," *Disposición Nro. 01-10. Ministerio de Educación*, 2010.

[2] S. M. Montgomery and L. N. Groat, "Student Learning Styles and their Implications for Teach-

ing," *CRLT Occasional Papers. Michigan University*, no. 10.

[3] K. Schneider, "Experience and Knowledge Management in Software Engineering," *Springer-Verlag Berlin*, 2009.

[4] A. Salama, *New Trends in Architectural Education: Designing the Design Studio.* Tailored Text & Unlimited Potential Publishing, 1995.

[5] S. Paavola, "Abduction Through Grammar, Critic, and Methodeutic.," *Transactions of the Charles S. Pierce Society: A Quarterly Journal in American Philosophy, 40(2), pp 245-270*, 2004.

[6] A. Russo and B. Nuseibeh, "On the Use of Logical Abduction in Software Engineering," *Handbook of Software Engineering and Knowledge Engineering, World Scientific*, pp. 889–914, 2001.

[7] Y. Wang, "The Cognitive Processes of Abstraction and Formal Inferences," *Fourth IEEE Conference on Cognitive Informatics, 2005. (ICCI 2005)*, pp. 18–26, 2005.

[8] F. G. Schuster, "Explicación y Predicción. La Validez del Conocimiento en Ciencias Sociales," *CLACSO. Argentina.*

[9] R. E. Nisbett and A. Norenzayan, "Culture and Cognition," *Chapter for D. L. Medin (Ed.) Steven's Handbook of Experimental Psychology.*

[10] K. Popper, "Logik der Forschung," *Verlag Von Julius Springer, Vienna, Austria.*

[11] J. V. Wertsch, "Vygotsky and the Social Formation of Mind," *Cambridge, MA: Hardvard University Press*, 1985.

[12] P. Denny, "Rational thought in oral culture and literate decontextualization.," *Chapter 4 In: Literacy and Orality, D. R. Olson and N. Torrance. Cambridge University Press.*, 1991.

[13] J. Allert, "Learning Style and Factors contributing to Sucess in an Introductory Computer Science Course," *Proceedings of the IEEE International Conference on Advanced Learning Technologies.*

[14] A. T. Chamillard and D. Karolick, "Using Learning Style Data in an Introductory Computer Science Course," *Proceedings of the thirtieth SIGCSE technical Symposium on Computer Science Education.*

[15] L. Thomas, M. Ratcliffe, J. Woodbury, and E. Jarman, "Learning Styles and Performance in the Introductory Programming Sequence," *Proceedings of the 33rd SIGSE Technical Symposium on Computer Science Education.*

[16] I. A. Zualkernan, J. Allert, and G. Qadah, "Learning Styles of Computer Programming Students: A Middle-Eastern and American Comparison," *IEEE Transactions on Education*, vol. 4, no. 49, pp. 443–450, 2006.

[17] A. F. Grasha, "Teaching with Style," *University of Cincinnati. Alliance Publishers*, 1996.

[18] L. Curry, "One Critique of the Research on Learning Styles," *Educational Leadership*, no. 48, pp. 50–56.

A. EXPERIMENT'S TASKS

A.1 DED-Task Sample

The following program produces one and only one of the three executions shown below:

```
def programD():
  n = input("Input an int.:")
  i = 0
  while (i<11):
    if (i%2 == 0):
      print i
    else:
      print i+n
    i = i+1
```

Your answer: Case 1, Case 2 or Case 3 ?.....

A.2 ABD-Task Sample

Just one of the three pieces of execution (the executions are partially shown) could belong to the program below:

```
def programA():
  first = 0
  second = 1
  print second
  for i in range(0,10):
```

case 1	case 2	case 3
Input an int.:1	Input an int.:2	Input an int.:3
0	0	0
1	3	4
1	2	2
2	5	6
2	4	4
4	7	8
4	6	6
6	9	10
6	8	8
8	10	12
8	11	10

case 1	case 2	case 3
...	...	...
2	2	2
3	3	3
4	5	10
...	...	...

```
next = first + second
print next
first = second
second = next
```

Your answer: Case 1, Case 2 or Case 3 ?.....

A.3 IND-Task Sample

The three executions (executions are completely shown) belong to one, and only one, of the programs shown below:

case 1	case 2	case 3
Input an int.:1	Input an int.:2	Input an int.:3
1	2	3
2	4	6
4	8	12
8	16	24
16	32	48
32	64	96
64	128	192
128	256	384
256	512	768
512	1024	1536

```
def progr1():
  n = input("input an int.:")
  j = 1
  f = n
  while (j<10):
    print f
    f = 2*j
    j = j+1

def progr2():
  n = input("input an int.:")
  j = 1
  f = n
  while (j<10):
    print f
    f = 2*f
    j = j+1

def progr3():
  n = input("input an int.:")
  j = 0
  f = n
  while (j<10):
    print f
    f = 2*j
    j = j+1
```

Your answer: Progr1, Progr2 or Progr3 ?.....

B Hand-on Activity Sample

In this section we include an instance of the software requirements given to the subject to develop a Python program using concepts of modularization, lists, and basic control structures.

Common structure of the statement for all the groups:

E-commerce grew by 49.5% in 2011 in Argentina. *...several real factors explaining the growth are listed...*

Usually the stock of products are recorded in a database and users making purchases are able to manipulate a list of products that works like a shopping cart. Once the final list of product is selected, the purchase is done through any means of payment (usually credit card).

Although we have not begun to program for the Web and its presentation layers (Web forms, pages, etc..) we are working with concepts that are useful to model the application domain layer of a system.

The program will contain two lists: a list that we call BDProducts (simulating what would be a database of a company) and another list named SelectionList (which will be a list of products selected by a buyer).

We will use items and products of the market to give the problem a better approximation to reality.

Variable part of the statement: (each group has different statements):

Your program should model a database of books with this structure (ISBN, Title, ApellidoAutor, NombreAutor, cantidadDePaginas, Year, Edition, Gender, valorUnitario, cantidadEjemplares). The base will be modeled using a Python list of tuples. The program will initially request a username and password to the user. Usernames "guest" with password "guest" are clients of the company. Other names of users will be administrators.

Administrators can perform the following tasks according to the following menu:

- Add Book to BDProducts (invokes a method that requests data from a book and adds it to the BDProduct, whenever the book does not exist)
- Get the number of different books of a particular genre (assume that there are only three general Science, BestSeller, Biography) and sprout the quantity of books of this genre.
- Get a list of books whose title contains a certain word.
- Get a list of books whose SurnameAuthor contains a particular surname.
- Get a list of all the books of the BDProducts.
- Delete a Book BDProductos (invokes a method that eliminates the book)
- Modify a Book description into the BDProducts (invokes a method that requests an ISBN and modifies data in that book BDProducts eg. modify its author name)
- Add (increase) a copy of a book into the BDProducts (invokes a method that receives an ISBN and increases the number of occurrences of that book in BDProducts)
- Decrement by one the number of copies of a book in the BDProducts (invokes a method that receives an ISBN and decrements the number of occurrences of that book in BDProducts)
- Get the number of copies of a book (given an ISBN)

Customers of the company should be able to: (1) List the books offered by the company; (2) List the company books available by gender; (3) Add to your list of selected products (SelectionList) a copy of a book. Initially the list is empty. Note that you can add more than one copy of the same book. (4) List the books on your list of selected products. Including a final price. (5) Remove a book from your list of selected products.