A NEW PARADIGM OF STUDY OF OBJECT ORIENTED PROGRAMMING: A COMPARATIVE APPROACH

***Pankaj K. Mudholkar, #Megha Mudholkar**

**Asst. Professor,Thakur Institute of Management Studies, Career Development & Research, Kandivali (E),*

#Lecturer,Thadomal Shahani Engineering College, Bandra (W),

ABSTRACT

A new approach of teaching object oriented programming languages is discussed in this paper. The approach introduced in this paper is specially designed for students who want to begin to learn object oriented programming. This papers discusses the comparative approach of teaching object oriented programming languages. There are various approaches of teaching object oriented programming languages. It was found that a new approach introduced in this paper found better impact on the knowledge gained by the students. This paper uses the technique of teaching VB.Net using comparative approach

General Terms: *Languages.*

Keywords: *OOP, VB, .Net, Teaching Path*

INTRODUCTION

Engineering and science are traditionally taught deductively. The instructor introduces a topic by lecturing on basic concepts of the topic and algorithms if necessary, then uses these algorithms to develop applications, shows illustrative applications, gives students practice in similar kind of applications in homework, and finally tests their ability to do the same sorts of things on exams. Little or no attention is initially paid to the question of why any of that is being done—what real world phenomena can the algorithms explain, what practical problems can they be used to solve, and why the students should care about any of it, what is the reason behind studying the programming language we are learning, why this programming language is, what are the basic differences between the other programming languages and the language we are learning. The only motivation to learn that students get—if they get any at all—is suggestions that the material will be important later in the curriculum or in their careers.

Most college students undergo a developmental progression from a belief in the certainty of knowledge and the omniscience of authorities to an acknowledgment of the uncertainty and contextual nature of knowledge, acceptance of personal responsibility for determining truth, inclination and ability to gather supporting evidence for judgments, and openness to change if new evidence is forthcoming [1,2]. Atthe

highest developmental level normally seen in college students (termed ‒contextual relativism‖ by Perry [1]), individuals display thinking patterns resembling those of expert scientists and engineers. A goal of science and engineering instruction should be to advance students to that level by the time they graduate.

In their courses, students may be inclined to approach learning in one of three ways[3]. Some take a surface approach, relying on rote memorization and mechanical formula substitution and making little or no effort to understand the material being taught. Others may adopt a deep approach, probing and questioning and exploring the limits of applicability of new material. Still others use a strategic approach, doing whatever is necessary to get the highest grade they can, taking a surface approach if that suffices and a deep approach whennecessary.

Another goal of instruction should be to induce students to adopt a deep approach to subjects that are important for their professional or personal development.

Almost all the students majored in computer science learn programming languages. The students must have to learn the basis on which the computer language they are learning is build. They must understand, the previous versions of the language available, differences between previous and current version, what‘s new in the current version, what are the benefits of the current version. For example, if we are teaching VB.Net to the students, start with the basics of Visual Basic (herein after VB) language although it is not in their curriculum, the differences in VB and VB.Net programming language. Since the grammars of VB.Net are very similar to VB language, particular emphases will be put on the concepts and technologies of OOP (Objected Oriented Programming)[4-6].

The students have learned the basics of programming, e.g., how to define and use function, how to define a variable, what a local or global variable is. However, all these concepts are the elements of procedure oriented programming like C language. So the students must have to learn how to use these concepts in object based programming e.g. VB language and then in turn in object oriented programming e.g. VB.Net. At the beginning, it is usually difficult for students to understand the underlying concepts of OOP. For example, they may easily know what an object is, what a class is, and how they are related. However, they don‘t understand the underlying ideas of defining a VB.Net class. They may doubt that why to define constructors and methods for a class. They even think that it is an example of making simple thingscomplicated.

The reason is simple. We put too much emphasis on the .Net language specification (such as grammars, types, variables, and so on), and the examples is not well designed and arranged. We just tell the students what a class looks like and how to define it through a few simple examples. However, we do not tell them why to define the members of a class, when to define these members and what the underlying ideas and principles are. If the students know the reasons, they will be more active to learn the .Net programming. Therefore, the teaching emphases should be the underlying thinking of OOP and the .Net features (here we are talking about VB.Net programming) relating to OOP[3-5].

In this paper, we will discuss various approaches (methods) of teaching object oriented programming languages. A Comparative approach is introduced in the paper, and the teaching path using this

approach is described by taking an example of teaching VB.Net. The study of three approaches i.e. linear, demonstration and comparative was done and the effective approach for teaching object oriented programming languages was found. Along this path, the concepts of VB.Net language was taught by comparing them with VB. The principles of object based programming languages (e.g. VB) and object oriented programming languages (e.g. VB.Net) will be taught step by step, and gradually introducing the students into the world of .Net OOP.

TEACHING APPROACHES AND TECHNIQUES

A teaching method comprises the principles and methods used for instruction. Commonly used teaching methods may include class participation, demonstration, recitation, memorization, or combinations of these. The choice of an appropriate teaching method depends largely on the information or skill that is being taught, and it may also be influenced by the aptitude and enthusiasm of the students.

Explaining, or lecturing, is the process of teaching by giving spoken explanations of the subject that is to be learned. Lecturing is often accompanied by visual aids to help students visualize an object or problem.

Linear Approach

Textbooks, curricula, and our educational system itself are the products of a mechanistic past. School knowledge is pre-determined by a centralized authority, and delivered in a linear format to a mass audience. The system is standardized, mass produced, scheduled, etc. In the classroom, the emphasis has been on teaching - it is expected that the learning will simply follow. The act of teaching, then, is seen as transferring information in a controlled sequence, a process that eliminates context - all learners receive the same content in the same format – but fails to accommodate variations in learnerneeds.

At the individual level, traditional learning is also _linear‘. Most textbooks stagger information – you can‘t proceed to Unit 2 until you‘ve learned Unit 1, type of thing. For example: Every book on programming language, have begun with the history of the language, data types, variables, arrays, operators, control statements and so on.. In fact, however, most learners of these book do not _acquire‘ the earliest items until they reach an advanced stage of programming. It‘s obvious that these sequence of items are presented out of expediency. The question is, however, whose expediency – the teachers‘ or the students‘? (There is no natural order of language learning that can be described as a linear set of morphemes.)

In linear approach, the subject is taught linearly, from start to end. Here we fail to accommodate learner‘s needs. We are teaching the subject linearly even if the learner knows everything. As an example, the students from various background taking admissions to MCA course. So the students who are from computer science background may know some programming languages they learnt which is there in the MCA curriculum. When we are teaching these kinds of subjects we must keep in mind the other students who do not knowanything.

Demonstration

Demonstrating is the process of teaching through examples or experiments. For example, a science teacher may teach an idea by performing an experiment for students. A demonstration may be used to prove a fact through a combination of visual evidence and associated reasoning.

Demonstrations are similar to written storytelling and examples in that they allow students to personally relate to the presented information. Memorization of a list of facts is a detached and impersonal experience, whereas the same information, conveyed through demonstration, becomes personally relatable. Demonstrations help to raise student interest and reinforce memory retention because they provide connections between facts and real-world applications of those facts. Lectures, on the other hand, are often geared more towards factual presentation than connectivelearning.

While teaching object oriented programming, generally we are using this concept. We will teach the basic concepts and will provide a demonstration on the projector either by writing the program, discussing the program and executing the program. This method of teaching provides good result and most of the faculties opting demonstration as a method of teaching.

Collaboration

Collaboration allows students to actively participate in the learning process by talking with each other and listening to other points of view. Collaboration establishes a personal connection between students and the topic of study and it helps students think in a less personally biased way. Group projects and discussions are examples of this teaching method. Teachers may employ collaboration to assess student's abilities to work as a team, leadership skills, or presentationabilities.

Collaborative learning is "an instruction method in which students at various performance levels work together in small groups toward a common goal. The students are responsible for one another's learning as well as their own. Thus, the success of one student helps other students to be successful."

Group project is allotted to the students and they have to develop an application using any programming language. The students will discuss the issues, key points, feasibility study of the application they are developing and errors if they are getting while developing the system.

A COMPARATIVE APPROACH OF TEACHING

The comparative method is a technique for studying the development of languages by performing a feature-by-feature comparison of two or more languages with common descent from a shared ancestor.

The Comparative Method as such is not, in fact, historical; it provides evidence of linguistic relationships to which we may give a historical interpretation. ...[Our increased knowledge about the historical processes involved] has probably made historical linguists less prone to equate the idealizations required by the method with historical reality Provided we keep [the interpretation ofthe results and the method itself] apart, the Comparative Method can continue to be used in the reconstruction of earlier stages of languages (Fox, 1995)

In this section, we discuss how to teach the basics of VB.Net programming to the students by using this approach. The teaching content of this part is not deeply involved with the techniques of OOP, and comparisons between VB and VB.Net language are often discussed.

The main teaching aim includes:

- Let the students know the distinctions between VB andVB.Net;
- Introduce some important concepts of VB.Netlanguage;
- Let the students grasp the ways and means of using the predefined classes ofVB.Net.

VB.NET is the most recent generation of Visual Basic. Developers will be pleased to note that its new features include inheritance, method overloading, structured exception handling, and more. These capabilities make it easier than ever to create .NET applications, including Windows applications, web services, and web applications. This article is mainly focused on a teaching path for VB.Net programming. This path is specially designed for learning of a VB.Net programming.

It was observed that, many of us who have already learned Visual Basic (herein after VB) and even not learned VB but wants to begin to learn VB.Net programming, they start the journey but they are not basically focused on why .Net is and What is the exact difference between VB and VB.Net. Along this path, to begin with .Net programming, the concepts and principles of object oriented programming must be understood step by step. We have to learn the specifications of the VB.Net as well as the underlying ideas of object oriented programming. We will gradually break through the obstacles between the object based programming and the object oriented programming.

At the beginning, it is usually difficult to understand the difference between VB and VB.Net. Why VB is object based and VB.Net is object oriented, what an object is, what a class is and how they are related, why to define constructors and methods for a class. However, we don‘t understand the basic differences of defining the variables in both languages, why and how this differs; we don‘t focus our attention on the data type differences in thelanguages.

A Simple Application Program

At the beginning of the course, we should give a few simple examples to show what a VB program looks like. The students must be aware of how to write programs in VB although it is not mandatory to learn VB before VB.Net. But for simplicity, the students must be aware of why VB.Net is and what additional features VB.Net provides over VB. Then we should give a few simple examples to show what a VB.Net program looks like. Since the students is familiar with VB language, i.e., they know how to write a program with a few functions, the selected examples should be like a VB program to make the studentssurethatVB.Netprogrammingisnotdifficult.Aprogram–HelloWorld‖isoftenusedasthe first example. The students can easily see the differences and similarities between VB.Net program and VBprogram.

Primitive Data Types

The grammars of VB.Net language are almost similar to VB language. It seems a good thing for the students with knowledge of VB programming. It is necessary to distinguish the differences of grammar and syntax between VB and VB.Net language. We must often remind the students that the VB.Net is just similar to (but not the same as) VB language. An important aspect regarding the data type is the .Net platform provides common type system to all supported languages. This means that all the languages must support the same data types as enforced by common language runtime. This eliminates data type incompabilities between various languages. For example, on the 32 bit platform, the windows platform, the integer data type take 4 bytes which is the same case with VB.Net whereas in VB takes 2 bytes. VB.Net does not have variant data type. To achieve the similar result to variant type you can use Object data type (Since everything in .Net including primitive data types is an object, a variable of object type can point to any data type). Since VB.NET uses a common runtime among Visual Studio languages, data types in VB needed to be brought into line with those of other languages. Here are some of the data type changes.

Table 1. Comparison of VB and VB.Net Data types

VB 6	VB.Net	Comments
Integer	Short	16 bits
Long	Integer	32 bits
N/A	Long	64 bits
Variant	N/A	Use the new _Object' data type
Currency	N/A	Use Decimal in VB 6 or Decimal or Long in VB.Net
N/A	Decimal	Available in VB 6. Native inVB.Net
String	String	VB.Net does not supports fixed length strings

VB.NET no longer supports the Variant and Currency data types. VB documentation has always warned to minimize your use of variants. In .Net they are not supported. Use the object data type instead.

The currency data type stores values as 64-bit integers, scaled by 10,000 to give a fixed-point number with 15 digits to the left of the decimal point and 4 digits to the right. In VB.NET you can replace currency variables with the new 64-bit Long variable or the 64-bit decimal data type.

In VB6, you can also use the decimal data type. Although you cannot declare a variable as decimal, you can declare it as variant then use the cDec function to make its subtype decimal.

Most APIs that take numeric arguments expect 32-bit values. In VB6 that's a Long data type. In VB.NET a long is 64-bits and will not work with 32-bit API calls. Your .NET API parameters will have to be changed or cast to the Integer data type prior to invoking theAPI.

Variable Declaration and Arrays

There are so many differences in declaring the variables in VB and VB.Net. In VB it is a good practice to declare variables on separate lines and it seems to be good practice anyway. For example, if variables are declaredas,

```
Dim a As Integer
Dim b As Integer
```

The above statements declare both variables as Integer in VB and VB.Net.

But if the variables declared as,

```
Dim a, b As Integer
```

In this example VB6 will consider a as variant and b as integer, which is somewhat odd behavior. VB.NET corrects this problem, creating both a and b as integers.

So we may say that VB permits multiple variable declarations in a single line, but only the last variable is defined as a particular data type and rests are variant whereas VB.Net corrects this problem and every variable is defined as a given data type, in the given example VB.Net will consider variables a and b both as aInteger.

In VB6 declaring an array declared as,

```
Dim Items (5) As String
```

gives you 6 items from index 0 to index 5. In VB.Net, this same declaration will yield 5 elements from index 0 through index 4. Be on the look out for "out of bounds" type errors. Also, all arrays in .NET must now bezero-based.

RESULT AND DISCUSSION

Data is collected from the batch of 10 students studying VB.Net using linear, demonstration and comparative approach. A questionnaire based on the general awareness of the language is distributed and the data is collected. The following table shows the percentage of students answering correctly by teaching them using linear, demonstration and comparativeapproach.

Table 2. Students performance using linear, demonstration and comparative approach

Teaching	Number of	Percentage

Methods	Students passed (30 students from eachcategory)	of Passing
Linear	4	13 %
Demonstration	10	33 %
Comparative	16	54 %

As shown in the table, it was found that the comparative approach of study is the better option for understanding the concepts as compared to linear and demonstration approach. The passing percentage of students taught by comparative approach is much greater that linear and demonstration approach.

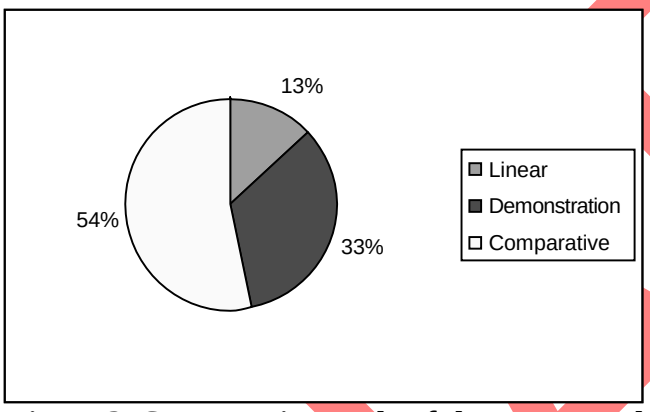

Figure 3. Comparative study of three approaches

Figure 1 shows the 54% of students cleared the exam using comparative approach, which is much greater than linear where only 13% students cleared the exam and demonstrative where 33% students cleared theexam.

CONCLUSION

We have discussed a teaching path for the students who have learned VB language and begin to learn VB.Net programming. At the beginning of the path, we allow the students to write a VB.Net program like a VB program. The emphasis of this phase is to make the students familiar with some importance concepts of VB.Net language. The second phase is to discuss how to define a class. In this phase, we emphasize the benefits of providing constructors, methods for a class, and why to control access to a class. The third phase is to discuss how to write a subclass. The meanings of methods overriding are particularly emphasized in this phase. Along this path, the concepts and principles of OOP are taught step by step. We don‘t just tell the students how, but also tell them why. The student can learn the

specification of .Net language as well as the underlying ideas. The students will break through the obstacles gradually between the object based programming and the object oriented programming.

REFERENCES

[1] Perry, W.G., Forms of Intellectual and Ethical Development in the College Years: A Scheme, San Francisco, Cal.: Jossey-Bass, 1988. (An updated reprint of the original 1970volume.)

[2] Felder,R.M.,and Brent,R.,–TheIntellectualDevelopmentofScienceandEngineeringStudents: Models and Challenges,‖ J. Engr. Education, Vol. 93, No. 4, 2004, pp. 69–277, <http://www.ncsu.edu/felder-public/Papers/IntDev-I.pdf>.

[3] Marton,F.andSäljö,R.,–ApproachestoLearning,‖inF.Marton,D.Hounsell,andN.Entwistle, eds., The Experience of Learning, 2nd ed., Edinburgh: Scottish Academic Press,1997.

[4] JeffreyR. Shapiro, (2009) –TheCompeteReference VisualBasic.Net‖, Tata McGraw Hill.

[5] SuJian,WengWenyong,WangZebing,(2009)–ATeaching Pathfor JavaObjectOrientedOriented Programming‖,IEEE.

[6] Evangelos Petroutsos, Mark Ridgeway, (2003) –Visual Basic .Net Developers Handbook‖, BPB Publication

[7] AnnePrince, (2003) –Murach‘sBeginningVisualBasic.Net‖,BPBPublication.

[8] Kris Jamsa, (2003) "Visual Basic .Net Tips and Techniques", Tata McGrawHill.

[9] Craig Utley, (2003) "A Programmer‘s Introduction to the Visual Basic .Net‖,Techmedia.

[10] Noel Jerke, (1999)–TheCompeteReferenceVisual Basic.Net‖,TataMcGrawHill.

[11] Steven Holzer, (1999)–Visual Basic Programming BlackBook‖, Coriolis.

[12] Dietel &Dietel, T. R. Nieto,(1999)–Visual BasicHowto Program‖,Pearson Education.

Fox, Anthony (1995). Linguistic Reconstruction: An Introduction to Theory and Method. New York: Oxford University Press.