Journal of Engineering Education Transformations ,
Volume 32 , No. 1, July 2018, ISSN 2349-2473, eISSN 2394-1707

A Case Study on Curiosity Point Based Teaching and Learning– A Step Towards Industry Readiness

U.P.Kulkarni [1], S.B.Kulkarni [2], S.M.Joshi [3], Indira R. Umarji[4]
[1, 2, 3, 4] Institutional Accreditation Unit, SDM College of Engineering and Technology, Dharwad, India.
[1] upkulkarni@yahoo.com
[2] sbkulkarni_in@yahoo.com
[3] joshshree@gmail.com
[4] indira.umarji@gmail.com

Abstract: The Outcome Based Education (OBE) being the core academic quality drive of all Engineering Colleges in India, particularly for tier-1 Institutions, yet industry readiness continues as a major matter of concern. Though the competence and performance of learner is stated clearly in terms of observable and measurable outcomes that are essential to carry out the role effectively by learners, yet the style of teaching and assessment by teachers to help the students' learning has great impact on achieving the specified outcomes. This style of teaching or teaching methodology differs from teacher to teacher. This variation or lack of maturity in understanding sufficiency and completeness as a parameter for attainment of outcome influences students' Industry readiness.

Authors of this paper present a case study on a concept called 'curiosity points', a measure of sufficiency and completeness of learning any topic. This case study focuses on computer programming at both first semester undergraduate level and at postgraduate level. The concept may be extended to any course of any branch.

Keywords: OBE, NBA, AICTE, Quality, Autonomous.

U.P.Kulkarni
Institutional Accreditation Unit,
SDM College of Engineering and Technology, Dharwad, India.
upkulkarni@yahoo.com

1. Introduction

The true understanding of OBE and tuning the current functioning of an educational institution is necessary for the success of higher educational policies of this country. Recent introduction of Outcome Based Education (OBE) has been partially understood and has created several gaps between “the truth” and what is being perceived [6].

Outcome-based education (OBE) is an educational theory focusing on following core principles [3]:

a) All students should emerge from the system as genuinely successful learners.

b) All students can learn and succeed, but not on the same day in the same way.

c) Successful learning promotes even more successful learning.

d) Stating clearly the competence and performance of learners in terms of observable and measurable outcomes that are essential to carry out the role effectively by learners.

e) Incorporating active action oriented words that reflect critical or higher order thinking into learning outcome statements.

f) Defining the style of teaching and assessment by teachers to help the students to achieve the specified outcomes.

g) Focusing on WHAT is to be learned and WHETHER it is learned well or not.

h) Viewing curriculum, instruction, and assessment as flexible and alterable means for accomplishing clearly defined learning "ends".

i) Encouraging the teachers to explore better ways of designing and delivering instruction, especially in light of differences in student learning rates and styles.

j) Using pencil grading rather than pen grading i.e. opportunity for students to truly catch up and erase the records of earlier mistakes.

Out of these many core principles of OBE, this paper focuses on major concern of style of teaching and its impact on student's learning outcome. This case study focuses on computer programming at both under graduate level and at post graduate level. The concept may be extended to any course of any branch.

2. Proposed Model

Programming languages are to be taught to every citizen as it increases the logical thinking and reasoning. If a student is successful in programming, then it means that he or she can learn any course independently.

However, the bad style of coverage of the programming course makes the students' learning fall short enough to meet for the industry readiness. Different levels of learning a programming language are listed below ordered from primitive level (1) to expertise level (4).

1. Syntax of a programming language- Primitive.

2. Problem solving- A logical reasoning.

3. Curiosity-points coverage focusing on principles.

4. Application development- Expertise Level.

Current Practice:

Most of the teachers focus on syntax of a programming language through simple illustration at level-1. However, the coverage at problem solving to enhance logical reasoning is neglected for lack of time as syllabus is filled with syntaxes of language with little scope on problem solving. This problem solving enhances the logical reasoning capability of the students.Curiosity points are such related indirect topics at principles level, which are not being addressed. The curiosity points transform the learning at sufficiency level (exam oriented learning) to completeness level (industry readiness). This is illustrated in the next section.

Improper learning style in lower semester particularly in programming language, data structure, algorithms and exposure to current technological trends and industry standards makes students lack in solving the problem that has some relevance in the society. This is quite visible in academic projects at higher semester levels where students lack following graduate attributes.

1. Ability to handle challenging issues.

2. Ability to manage large scale.

3. Ability to show interpersonal communication skills.

4. Ability to solve real problems, not just write code and move bits.

5. Ability to design and improve the systems based on a quantitative and qualitative assessment of its functionality, usability and performance.

6. Ability to communicate their solution to others, including why and how a solution solves the problem and what assumptions were made.

7. Ability to interplay between theory and practice.

8. Ability to manage their own learning and development, including managing time, priorities, and progress.

The main crux of complete learning is the coverage of 'CURIOSITY POINTS' and the same is described in the next section.

Case Study:

To make learning complete towards industry readiness, a course teacher needs to focus on coverage of all dependent topics called as 'Curiosity Points'. The

coverage of curiosity points at principles of programming language makes subsequent learning also more successful. This is shown in Fig-1.

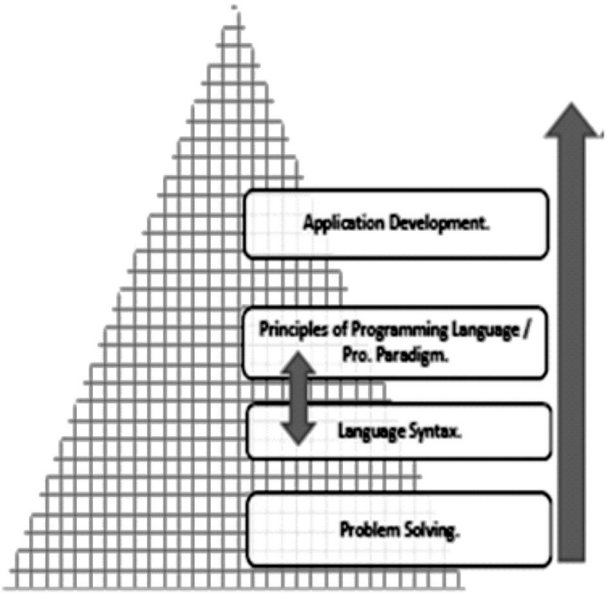

Fig. 1Different aspects of Learning Programming Language.

Following are the few cases that highlight the class roomcoverage at 1st semester level on the topic of writingprograms in C language

Case-1:

What is the output of this program?

```
1.  char a;
2.  for (a=0; a<=255; a++)
3.  {
4.      printf ("\n %d \n",a);
5.  }
```

It prints value from 0 to 127 and then -128 to -1 and repeats this sequence continuously.

The question is why? This is being referred as curiosity points and are listed below.

C1. Range of numbers represented by char data type.

C2. After printing 127 why it becomes -128?

The explanation for these two makes learning a complete one. The explanation goes as below:

C1: Character data type is represented as 1 byte and its range of values is from -128 to 127.

C2: When the loop takes value of variable 'a' to 127, an increment operation makes it -128. The explanation for this by example, as shown in table-1 and Table-2 makes students' learning more successful. This successful learning makes students subsequent learning much better. Teachers are expected to explain this concept by taking students to another connected topic called binary representation of a number and its arithmetic. If students have not learnt this in earlier level properly or teacher skip this explanation, then students fail to reason out why it is so.

Table-1: Explanation for how incrementing 127 becomes -128

Bit Position - n; 8 bit for a character in C language.	7	6	6	4	3	2	1	0
Position Value - 2n	128	64	32	16	8	4	2	1
a=127	0	1	1	1	1	1	1	1
When 127 is incremented, sign bit (7) becomes 1 indicating negative number.	1	0	0	0	0	0	0	0
Since MSB bit position(7) is 1, it is negative number and 2's complement of this is -128	1	0	0	0	0	0	0	0
When this number (-128) further incremented by 1	1	0	0	0	0	0	0	1
Its value in 2's compliment is -127	1	1	1	1	1	1	1	1

Table-2: Explanation for how incrementing -1 becomes 0

Bit Position-n	7	6	6	4	3	2	1	0
Position Value - 2n	128	64	32	16	8	4	2	1
a= -1	1	1	1	1	1	1	1	1
Incrementing a makes it 0	0	0	0	0	0	0	0	0

Case-2:

What type of parameter passing technique exist in CLanguage?

While teaching parameter passing technique at first semester level, the emphasis on various concepts of parameter passing principles and its existence in a given language is not given. This makes learning incomplete.

C language supports only call by value. This is to be illustrated through simple example as given below.

```
1.  #include <stdio.h>
2.  void increment1 (int a);     // for call by value.
3.  void increment2(int *a); // for use of pointers in
4.                                  // parameter passing
5.  main()
6.  {
7.    int b = 20;         // automatic variable
8.   // Call by value Parameter passing technique
9.      printf("\n\n Before Calling: Value of b %d\n\n",b);
10. printf("\n\n Before Calling: Addres of b is : %d\n\n",&b);
11. //increment1(b);  // remove the comment and run the
12.                          // program
13. //increment2(&b); );  // remove the comment and run
14.                          // the program
15.
16. printf("\n\n After Calling: Value of b %d\n\n",b);
17. printf("\n\n After Calling: Address of b is : %d\n\n",&b);
18. }
19. // Procedure
20. void increment1(int b)
21. {
22.   b = b+10;
23.   printf("\n\n INSIDE Increment1: Address of b is :
24.   %d\n\n",&b);
25. }
26. void increment2(int *a)
27. {
28.  *a = *a +10;
29.   printf("\n\n INSIDE Increment2: Address of a is :
30.   %d\n\n",&a);
31.   printf("\n\n INSIDE Increment2: Content of a is :
32.   %d\n\n",a);
33.     printf("\n\n INSIDE Increment2: Content of
34.                  Memory pointed by  a is : %d\n\n",*a);
35. }
```

The output of above program is given below.

Test run with increment1 () function.

1. Before Calling: Value of b 20
2. Before Calling: Address of b is : 37814044
3. INSIDE Increment1: Address of b is : 37814008
4. After Calling: Value of b 20
5. After Calling: Address of b is : 37814044

Line 2 & 3 indicate that the actual parameter and formal parameter represent the different memory location and hence it supports the principle of call by value parameter.

Test run with increment2 () function.

1. Before Calling: Value of b 20
2. Before Calling: Address of b is : 37814044
3. INSIDE Increment2: Address of a is : 37814008
4. INSIDE Increment2: Content of a is : 37814044
5. INSIDE Increment2: Content of Memory pointed by a is : 30
6. After Calling: Value of b 30
7. After Calling: Address of b is : 37814044

In this run also we see that, Line 2 & 3 indicate that the actual parameter and formal parameter represent the different memory location and hence it support the principles of call by value parameter. However, since the variable 'a' is the pointer type variable containing the address of variable 'b', any manipulation through 'a' will affect the variable 'b' and hence change in the variable 'b' is seen in the step 6.

Case-3:

What is the output of the following code segment?

```
1. #include <stdio.h>
2. void fun();
3. main()
4. {
5. fun();
6. fun();
7. }
8. void fun()
9. {
        autointi;
        register int j=0;
        static int k =0;
        i++; j++; k++;
        printf("\n\n %d  %d  %d\n",i,j,k);
10. }
```

This code produces the output shown below.

1. 37814013(garbage) 1 1
2. 37814014(garbage) 1 2

This is because the initial value and scope of different storage classes as shown below.

Table-3: Details of different storage classes

Class	Storage	Default Value	Scope	Life Time / Scope
Automatic	Memory	Garbage	Block	Till Control is in block
Register	CPU Register	Garbage	Block	Till Control is in block
Static	Memory	Zero	Block	Persists across calls
External	Memory	Zero	Global	As long as Program is Running

Most of the classroom coverage don't focus on such simple concepts (Curiosity Points) and makes students' learning incomplete.

Case-4:

How memory is allocated for the variables used in the following code segment?

```
1. struct emp
2. {
     int a;
     char c;
     float f;
3. };
4. struct emp e;
5. printf("\n\n ADDRESS: %u %u %u\n",
     &e.a,&e.c,&e.f);
6. printf("\n Size is : %d\n", sizeof(e) );
```

Assuming 32 bit word size of a machine, the above code generate the following output.

1. ADDRESS: 37813920 37813924 37813928

2. Size is : 12

This is becausethere is a hole of 3bytes after the char and hence total size of structure is 12 bytes.

Similarly, students may be give a self-learning based exercise to motivate or trigger curiosity in enjoying the learning experience and hence driving lifelong learning characteristic of a graduate from 1st semester itself.

Self-learning Exercise:

What is the output of the following code segment?

```
struct emp
{
        int a:4;
        int b:4;
};
struct emp2
{
        int a;
        int b;
};
struct emp e;
struct emp2 e2;
printf("\n\n SIZE of Structure %d\n",sizeof(e));
printf("\n\n SIZE of Structure %d\n",sizeof(e2));
```

There are many such situations in various course; if covered byconnecting various other information called 'Curiosity Points' ,makes students' learning successful and thereby enhancingattainment of outcome and hence the industry readiness.

Few such situations related to programming paradigm are highlighted below.

1. Why size of integer in Java is 4 bytes?
2. How to write robust code?
3. What is the size of integer in C?
4. What are the features specified in C11 standard?
5. Impact of host byte ordering and network byte ordering?
6. What is the use of abstract class in the design?
7. What is the use of private inheritance?
8. How do you implement multiple inheritance in Java?
9. How abstract class is different from Interface in Java language.

3. Conclusion

From OBE perspective, it is very important to focus on style of teaching and assessment by teachers that help the students to achieve the specified outcomes. The variation or lack of maturity in understanding sufficiency and completeness in learning; referred here as 'curiosity points' influences students Industry readiness.

Acknowledgement

Authors of this paper sincerely acknowledge undergraduate and postgraduate students who have undergone training on this perspective.

References

[1] Rangan Banerjee, Vinayak P. Muley, "Engineering Education in India" draft final report sponsored by Observer Research Foundation, Energy Systems Engineering, IIT Bombay.

[2] Prof. Yeshpal, "Report of the committee to advice on renovation and rejuvenation of higher education" published in AICTE web site.

[3] William G. SPADY , "Outcome-Based Education: Critical Issues and Answer", Published by US Department of Education, American Association of School Administrators.ED380910, 1994

[4] Writing and Assessing Course-Level Student Learning Outcomes published by Office of Planning and Assessment, Texas Tech University.

[5] U.P.Kulkarni et.al, "Innovative Workflow for Accreditation Process to Ensure Quality Engineering Education in India", IEEE International Conference on Teaching, Assessment and Learning for Engineering 2013.

[6] U.P.Kulkarni et.al, "A Case Study on an Unrealized Truth of an Outcome Based Education and Corrective Strategies", 4th IEEE International Conference on MOOCs, Innovation and Technology in Education -2016.