January 2013

Difficulties in understanding object oriented programming concepts

Soly Mathew Biju
University of Wollongong, soly@uow.edu.au

Difficulties in understanding Object Oriented Programming Concepts

Abstract- *Understanding object oriented concepts is always a difficult task for students. It is equally challenging for lecturers to teach these concepts. Over the years teachers have used various methods to teach these concept. The result of a class test was analysed to identify the various areas students have difficulty in understanding. The result will help in designing course material that would focus on these areas of object oriented programming.*

Keywords: Object Oriented concepts, constructors, classes , data hiding , student centred learning, C++.

I. INTRODUCTION

Existing research shows that there have been a number of problems faced by teachers teaching programming to undergraduate students. Students find it very difficult to understand object oriented concepts like classes, constructor invocation, overloaded constructors, friend functions and other object oriented concepts [2].Students who have been exposed to procedural programming find it a little difficult to move towards object oriented programming. It takes some time for them to understand Object Oriented concepts. Object oriented concepts consist of a number of classes within a single program.

This paper discusses the concepts of object oriented programming which are difficult for students to understand.

II. TEACHING METHODS DEPLOYED

Student centred learning methods for programming classes are very effective. Student-centred learning consists of facilitating understanding and conceptual change or intellectual development [9]. As a teaching tool, programming assignments are constructed to encourage students' development of analytical, programming writing skills. Watkins [8] argues that a student/learning-centred concept of teaching is one where high quality learning is viewed as "requiring active construction of meaning and the possibility of conceptual change on the part of the learners".

In case of programming languages, this approach focuses on teaching concepts by encouraging students to write and implement a program based on a concept to solve the given problem.

Learning can be made possible within the lecture by making students do things which are called 'active learning' [11]. For example, while teaching the concept of classes, one of the examples used in an introductory class is as follows

```
class distance1
{
private:
        int feet;
        float inches;

public:
        void getdata();

        void showdata();
        };
void distance1:: getdata()
        {
                cout<<"Enter the feet "<<endl;
                    cin>>feet;
                    cout<<"enter the inches
"<<endl;
                    cin>>inches;
        }

        void distance1::showdata()
        {cout<<feet<<"\'-
"<<inches<<"\""<<endl;
        }

int main()
{   distance1 d1;
        d1.getdata();          //calling member
functions of the class with
        d1.showdata();//member access operator
        system ("pause");

        return 0;
}
```

The concept of encapsulation is taught. The fact that getdata () and showdata() are member functions of the class and therefore could access all the variable members is clearly explained to the students

When a program was given to the students to work on many of them questioned how the functions could access the variable members without passing them as parameters to the functions.
Though the students claimed to understand the concept of data encapsulation and the fact that the functions of a class can access the data members of that class, they do not really understand it unless they start writing codes based on that concept. To help them understand, these students are given home works and assignments based on the concept taught in the class.

Ramsden's [7] theory of Teaching as making learning possible seems to be appropriate in case of programming classes. Teaching is comprehended as a process of working cooperatively with learners to help them change their understanding. He also states that teaching involves finding out about students' misunderstandings, intervening to change them and create an environment for learning. In programming classes conducted by me, this is implemented by conducting quizzes and tests and providing appropriate feedback. Sometimes these tests bring to light certain concepts that have been misunderstood, in such cases, I make it a point to dedicating a part of the next session to reinstate the concept correctly.
Another method implemented by me, is using pictorial representation to describe different concepts or even use Unified Modelling Language diagrams wherever applicable.

III. THE RESEARCH

A research was conducted in order to identify the basic object oriented concepts students have difficulty in understanding in a programming course. The students in the course have a basic procedural programming background and this course is an introduction to object oriented concepts.

A. Research Questions

The research questions were:
1. To what extent do students understand the underlying process which takes place when creating an object, specifically when the constructors are overloaded?
2. To what extent do students understand how and which constructors are implicitly called when a parameter is passed which objects construction.
3. To what extent do students understand how copy constructors are used?
4. To what extent do students understand how friend constructors are called in a program?
5. Do the students understand the concept of data hiding and data encapsulation?
6. Do the students understand the working of a friend function?

B. Research Population

The research population had 30 students who took the test in the
course "Applied programming " in C++ in the summer semester in 2008 and2009 after some basic concepts of OOP were introduced.

C. Research Instruments

The research instrument was the class test which consisted of six questions covering one of the introduction chapters. The questions check whether the students have understood the concepts of data encapsulation, data hiding, constructors, and friend functions.
The test questions are as given below.

```
class A
{
private :
int n=0;
public:
 A(){
this->n=0;
cout<<"A constructor1";
}
A(int n){
this->n=n;
cout<<"A constructor2";
}
A(A &Obj){
this->n=Obj.n;
cout<<"A constructor3";
}
```

1.What will be printed as a result of the execution of the following statement?

```
   A a;
```

2. What will be printed as a result of the execution of the following statement?

```
    A b(9);
```

3. What will be printed as a result of the execution of the following statement?
A b(a); where a is an object of the class A that has already been created.

```
using namespace std;
class Student
{
        int Studentid;
        string StudentName;
        double marks[5];
        double average;
        static int count;
public:
        Student()
        {
                Studentid=0;
                StudentName="";
                average=0.0;
                count++;
        }
        void calAvg();
        void getInfo();
        void displayInfo();
        static int get_count()
        {
                return count;
        }
        friend                  double
highestAvg(Student[], int);
        friend                  void
highestDetail(Student[], int);
};
 int main()
{
        Student *stdArr;
        int count;
        cout<<"How many students do
you wish to add: ";
        cin>>count;
        stdArr=new Student[count];
        for (int i=0;i<count;i++)
        {
                stdArr[i].getInfo();
                stdArr[i].calAvg();
        }

-----------------------
----------------------
```

4. Add necessary lines of code to call the friend functions.

5. Consider the class student given above

```
1. int main()
2. {
3. Student s;
4. s.id=99;
5. s.getInfo();
6. }
```

which line will cause an error while compiling ?why?

6. Is there an error in the friend function of the class Student defined below?

If so correct it.

```
1. void display (Student s)
2. {
3. cout<<average;
4. }
```

IV. DISCUSSION

The result of the test is given below

Table 1 Test result

Section	Mean	STDEV
1	70	0.48305
2	70	0.48305
3	60	0.5164
4	70	0.48305
5	80	0.42164
6	70	0.48305

In question 1, the students were required to exhibit understanding of the underlying process of creating an object. The constructor invoked by the object 'a' created in this question will initialize n to 0 and prints A constructor1.

From table 1 we can see that 30% of the students did not thoroughly understand the process of implicit innovation of constructors. These students were confused and thought that the question had an error.

In question 2 the students were required to exhibit understanding of overriding of constructors.

The match is found with the second constructor with signature A(int).The output would be A constructor 2.

From the result above we can see that 30% of the students did not understand this concept. They wrote down both A constructor1 and A constructor2 in the answer sheet.

In question 3, the students were supposed to exhibit understanding of a constructor with the signature A(&Obj) that is the copy constructor is called.

We can see that 40 % of the students did not understand this.

In question 4, students were supposed to give the correct syntax for calling a friend function. A friend function does not belong to any class but is the friend of the class hence can access the

data members of the class. A friend function can access data members of the class only using the dot operator.
30% of the students have not understood the concept of friend functions.

In question 5, students were supposed to exhibit their understanding of data encapsulation. The data member *id* is private hence cannot be accessed from the *main()*. 20 % of the students did not understand this concept.
Question 6 is also related to friend function. the friend function accepts a parameter of the object *s* of class *student*.
30% of the students did not understand the concept of writing friend functions.

V. CONCLUSION AND FUTURE RESEARCH

To conclude, it is apparent from the above results that the students have a good understanding of the following concepts:

- Data hiding can be done by declaring the data member as private.A private data member cannot be accessed by functions outside the class.

On the other hand, the students had more difficulty with understanding the following concepts:
The benefits and use of friend functions were not clearly understood by the students.
The use of constructors and how overloaded constructors are invoked implicitly as soon as the object is created.

Overall, the above results are relatively good. Students were expected to do better and exhibit a better understanding of these concepts. The concept of encapsulation, classes and objects are little difficult for students to follow. This paper points out specific areas of difficulty which educators must be aware of so that they can plan the learning process accordingly.
Many researcher in this area believe in introductory level of programming, visualization for presenting new concepts for the students [3].
Academicians feel that there is a need to support the learning of OOP concepts using software tools [4],([5],[6][11]).
As a result of this study, I am planning to propose a change in the way the subject is taught.
For this subject, I plan to use the software Alice to introduce students to object oriented concepts. Alice is an Open education Resource made available to the teaching community by Carnegie Mellon. Alice was aimed at teaching computer programming to students in an interesting 3D environment.

Alice is used for teaching students to program rather than teaching students a specific programming language. It is a teaching tool for introductory computing. It uses 3D graphics and a drag-and-drop interface to facilitate a more engaging, less frustrating first programming experience [12].
Animated views can help the students in three central learning activities: Understand programs; Evaluate existing programs; Develop new programs [1].
In Alice's interactive interface, students drag and drop graphic tiles to create a program, where the instructions correspond to standard statements in a production oriented programming language, such as Java, C++, and C#. Alice allows students to understand Object Oriented Concepts better through animation programs. By manipulating the objects in their virtual world, students gain experience with all the programming constructs typically taught in an introductory programming course.
A study will be conducted to measure the understanding of object oriented concepts after the student have been introduced to Alice.

REFERENCES

[1] Stasko, J., Tango:,"A Framework and System for Algorithm Animation." *IEEE Computer*, 23(9), pp 27-39, 1990.
[2] Holland, S., Griffiths, R., Woodman, M.," Avoiding Object Misconceptions." in Proceedings of the 28th SIGCSE, pp. 131–134 (1997)
[3] E. Lahtinen and T. Ahoniemi, "Visualizations to support different levels of cognitive development," *Proceedings of the Fifth Finnish/Baltic Sea Conference on Computer Science Education*, November 2005.
[4] Ragonis, N., Ben-Ari, M.: On Understanding the Static's and Dynamics of Object-Oriented Programs. In: ACM SIGCSE, pp. 226–230 (2005)
[5] Murray, K.A., Heines, J.M., Kolling, M., Moore, T., Wagner, P.j., Schaller, N.C., Trono, J.A.:"Experiences with IDEs and Java Teaching: What Works and What Doesn't. "ACM SIGCSE Bulletin 35(3), 215–216 (2003)
[6] Roberts, E."An Overview of MiniJava", ACM SIGCSE Bulletin 33(1), 1–5 (2001).
[7] Martin, E., & Ramsden, P. (1993)," An expanding awareness: How lecturers change their understanding of teaching", *Research and Development in Higher Education, 15*, 148-155.
[8] Watkins, D. (1998). A cross-cultural look at perceptions of good teaching: Asia and the West. In J. J. F. Forest (Ed.), *University teaching: International perspectives* (pp. 19-34). New York: Garland Publishing Inc.
[9] Devlin, M. 2006,". Challenging Accepted Wisdom about the Place of Conceptions of Teaching in University Teaching", Improvement *International Journal of Teaching*

and Learning in Higher Education 18(2), 112-119 http://www.isetl.org/ijtlhe/ ISSN 1812-9129.
[10] Race, P. & Brown,S.(1998) ,"Refreshing your lecturing",The Lecturer's Toolkit. Kogan Page ltd, 19-49
[11] Ramsden ,P. (2003).,"Theory of teaching in higher education. Learning to Teach in Higher Education" , 2nd edn,RoutledgeFalmer, London, 106-116.
[12] Alice (1999), An educational software that teaches students computer programming in 3D environment. http://www.alice.or