Designing and Teaching of Language Independent Object Oriented Programming Course

Moula Husain [1], Mallikarjun S. Akki [2], Meena S. Maralappanavar [3], Narayan D. G. [4]
B.V.B College of Engineering and Technology,
Hubli, India,
moulahusain@bvb.edu [1], mallikarjun.akki@bvb.edu [2], msm@bvb.edu [3], narayan_dg@bvb.edu [4].

***Abstract*—Programming Languages play a crucial role in developing any software applications. Academicians and IT professionals give more priority for mastering many programming languages. Learning many language syntax and constructs will not contribute much in building problem solving skills. For any given task, students should be capable of solving it by using any of the programming languages. More importantly, students should build an art of optimizing the performance of the application by applying fundamental and advanced concepts of programming languages.**

In this paper, we propose an innovative way of designing and teaching of Object Oriented Programming Concepts without sticking to one particular language. In our first attempt, we practiced teaching of this course with two languages namely, C++ and Java. Our experimental results and students feedback prove the effectiveness of designing and teaching of Language independent Object Oriented Programming course. Moreover, it develops the skills of implementing and applying object oriented concepts using any object programming language.

Keywords—Bloom's Taxonomy; Optimization; abet a-k; Bridge Course;

I. Introduction

There is a tremendous growth in the generation and application of programming languages in the field of Computer and Information Technology (ICT) fields. Figure 1 shows top 10 programming popular languages used by the programmer around the world according to the survey conducted by IEEE Spectrum in collaboration with computational journalist Nick Diakopoulos. After knowing the number of programming languages, an important question will delve into every one's mind that how many languages a computer engineer should excel in. Every programming language has got its own importance and applications.

Many of the universities offer one structured programming language (preferably C), an Object Oriented programming Language (preferably C++) and a couple of Elective courses on other programming languages. In most of the current curriculum and teaching methodologies, Object Oriented Programming course is designed and taught with reference to one particular language either C++ or Java. Object Oriented Programming [2][3] provides benefits of data security, a great amount of reusability and software extensibility. It adopts bottom up approach for building an application and plays a critical role in optimizing any software application.

Figure1. Top 10 popular object oriented programming languages of the year 2014 – survey results from IEEE Spectrum working with computational journalist Nick Diakopoulos [1].

In this paper we propose a new approach of designing and teaching of Object Oriented Programming concepts without adhering to one particular language. Object Oriented programming concepts are given more importance rather than mere programming language syntax or constructs. Simultaneously we teach implementation of object oriented concepts by giving comparative analysis in more than one language. Through this, students will gain understanding of OOP concepts and at same time, make comparative study of implementation in more than one language.

In the previous work [4], an innovative methodology was adopted in Object Oriented Laboratory experiments where students build a small scale to medium scale game application in a span of one semester. Similarly in paper [5], Object Oriented Programming laboratory experiments are designed by using cooperative learning method based on game design and Visual Object Oriented Environment. An innovative homemade project based learning was adopted for teaching OOP concepts by Yen-Lin Chen et al.[6]. BlueJ Integrated Development Environment and Web based resources were used to teach OOP concepts in [7][8] by Kujis. In most of the previous work, researches were made on teaching OOP concepts using an interactive development tool, redesigning project assignments or laboratory experiments using game applications. But, our major contribution in this work is, redesigning of OOP course curriculum, so that object oriented concepts can be taught irrespective of language choice. Here course instructor gets an option to teach OOP subject according to his convenience and students learning. Also we provided an additional platform in the form of self study where in course instructor will teach OOP concepts in any one object oriented programming language and the other language will be given as self study component. As this course is associated with laboratory, implementation and application of OOP concepts may need to be carried out either in C++ or JAVA programming languages.

The paper is organized as follows. Next section describes curriculum design. In section 3 we discuss teaching methodology adopted in this course. Section 4 describes the assessment methods followed in the course. In final sections Results are demonstrated followed by conclusion and future scope.

II. Curriculum Design

In this section, we describe the Course Learning Objectives and content of the syllabus designed for the OOP course. Also we give comparison of previous and current curriculum.

A. Course Learning Objectives (CLOs)

The course learning objectives set for this course are as follows.

TABLE I Course Learning objectives

CLO No.	Course Learning Objectives
1	Explain usage of control structures, modularity, I/O. and other standard language constructs.
2	Apply the concepts of object oriented programming to develop a solution for an identified problem
3	Apply the concepts of object oriented design process.
4	Apply suitable exception handlers to make an application more reliable and robust.
5	Design graphical user interface for a given application.

Table I shows CLOs set for the new Object Oriented Programming course. First CLO describes about basic programming language features, second CLO explains about the implementation and application of OOP concepts to a real world application. Third CLO deals with adopting object oriented design process in developing software application. Fourth CLO is about using exception handlers and final CLO is about designing Graphical User Interface (GUI). Only first CLO is language specific which will be selected by the course instructor but rest of the CLOs can be addressed by using any of the Object Oriented Programming language.

B. Course content and comparison with previous course content

TABLE II Comparison between previous and current course contents

Ch.	Previous	Current
1	Introduction to Basic Language Features	Introduction to Basic Language Features
2	Classes and objects	Classes and objects, Encapsulation.
3	Stream Handling	Inheritance
4	Dynamic Memory Management	Object Oriented Design Process -UML diagrams
5	Inheritance	Polymorphism – static and dynamic polymorphism
6	Operator Overloading	Exception Handling
7	Dynamic Polymorphism	GUI and Event Driven Programming
8	Templates and Exception Handling	Introduction to Collection/ container classes.

From the Table II it can be clearly seen that, previous course content was designed with respect to particular language. It was more of language oriented rather than an object oriented course. Key features of any object oriented programming languages are 1) Classes and object 2) Encapsulation 3) Inheritance 4) Polymorphism 5) Exception Handling. Based on these key features we designed course chapters. More important point to note here that, these features are common in all object oriented programming languages. This course content is designed in such a way that it can be taught by using any of the object oriented programming language.

III. Teaching methodology

This course is of 5 credits course L-T-P-S: 4-0-0-1, where Lecture (L) – 4 credits, Tutorial (T) - 0 credit, Practical (P) – 0 credit and Self study(S) – 1 credit. We teach 4 credits object oriented programming course in any one language. Other language will be given as self study to the students. Self learning component is tested through theory exam questions, project activity and laboratory assignments. By this, we ensure that students will concentrate more on learning OOP concepts and be able to correlate implementation of concepts in more than languages.

The course being taught will cover the basic object oriented programming concepts. Where as in the class while teaching classes and objects, Unified Modeling Language

(UML) will be used through which a particular example like car or animals or bank account can be taken and the UML / class diagrams for the same are explained. Whereas in the lab the students are made to convert, the respective class diagrams into classes. Construction of the diagram requires the knowledge of object oriented concepts to design a solution with different classes with their data members, member functions and the relation between the different classes and its members. The UML diagram for Hospital Management System (HMS) designed by our students is shown in Figure 2. This shows the understanding of the concepts of the object oriented programming language (CLO - 2).

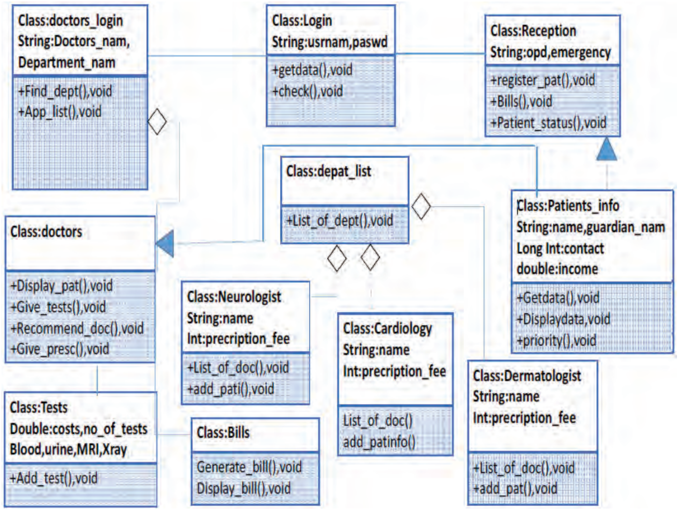

Figure 2. UML diagram for HMS.

To design the solution the students should understand the problem and the requirements completely. To build the designing skills in students, course project is introduced, wherein which students are applying their object oriented programming knowledge to design the solution. This approach of teaching helps to meet the CLO - 3. After this process the students analyze their design with the various scenarios to verify and validate the design, this process is necessary to prove the correctness of the design. Once they analyze the design the situation where the design is not meeting the requirements is identified. Then students are made to change the design with appropriate exception handling techniques to be used to code the system. This process makes the students to achieve the CLO – 4.

The Graphical User Interface (GUI) plays an important role in the system. It is one of the quality measurement components of the software system. Graphical interface provides Higher Productivity and Greater Accessibility for the software applications. The graphical interface for Hospital Management System designed by the students is shown in Figure 3(a). This process of selecting an appropriate GUI component to design the system is very important. This can be taught by explain the components and its properties, wherein students are asked list the scenarios where these components are mandatory. This assignment leads to CLO – 5.

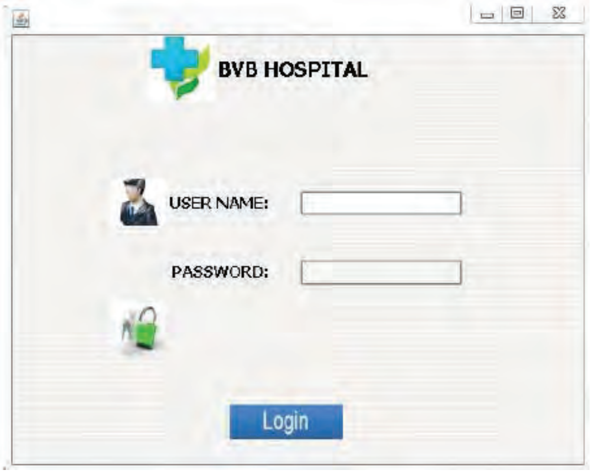

Figure 3(a). UI design for HMS.

The login screen of the Hospital Management System is which is used for authorization of the users is shown in the figure 3(a).

IV. Analysis

Course Articulation Matrix mapping of Course Learning Objectives (CLO) with Abet a-k [7] Criterion (Program outcomes). In achieving program outcomes, we have defined degree of compliance L: Low, M: Medium, H: High, based on the student outcome. As exist ABET criterion for program outcomes a-k.

TABLE III CLO's MAPPED WITH PO's

CLO no	PO Addressed	Degree of compliance (old)	Degree of compliance (new)
CLO1	a	M	M
CLO2	a,b	L	M
CLO3	a,c,d,g	L,L,L,L	L, H, L, L
CLO4	a	L	M
CLO5	a	L	M

By analyzing PO's, CLO's, degree of compliance and students outcome we have derived results for different CLO's. Comparing conventional or previously defined curriculum with current curriculum we witness drastic change in achieving PO's. Conventional methods were mainly focusing on the language we used like C++ or java but this strategy helps students to understand the important concepts of OOP in better way.

Course articulation matrix provides us with comparative result analysis of conventional and current course contents. Conventional methodologies were unable to achieve PO's with higher degrees, so redesigning of language independent learning of object oriented programming concepts does exist. The effectiveness of language independent learning of OOP concepts helps students to apply those concepts in their higher semester subjects like Java and C sharp etc. Focusing towards concepts rather than focusing towards language will make things general, students can apply these concepts in their course as well as projects of academics.

Distribution of marks weightage in terms percentage and mapping to corresponding levels expected are shown in Table IV.

TABLE IV Scheme of evaluation.

Evaluation Criteria	Weightage in %	Blooms level	PI Code
Formulation of problem statement	10	L3	b-1-A
Design and analysis of defined problem statement.	10	L3	b-2-B
Integrating developed modules for the defined problem.	50	L4	c-3-A c-3-B c-3-C
Test the modules.	20	L4	e-3-A
Technical report and presentation.	10	L2	g-1-B g-1-C

Student's performance in the semester exam indicates the increasing in the efficiency of the teaching approach used. Comparative analysis of the result of 2012-13 batch and 2013-14 batch is shown in the Figure 4. We can see that the increasing number in students getting higher grade in the examinations. In the academic year 2012-13 students getting the B+ grade was 72(S grade 12, A grade 24 and B grade 36) and in the year 2013-14 it is increased to 86 (S grade 16, A grade 26 and B grade 44).

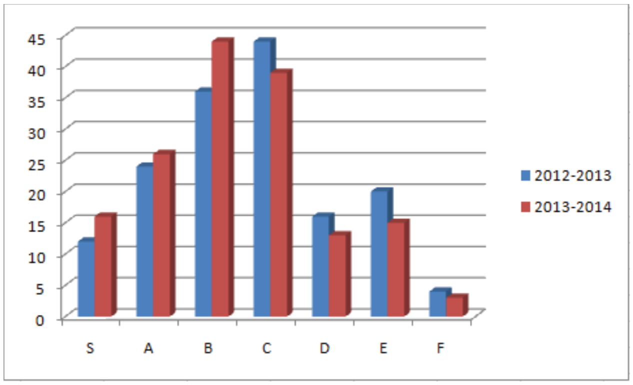

Figure 4. Analysis results.

V. Conclusion

This approach of teaching language independent OOP curriculum can effectively provoke student interests in applying OOP skills as well as cultivate their problem solving abilities, thus providing effective chance to learn the advantages of OOP. The students can widely and effectively adapt their learned concepts and skills for many practical aspects of student's research studies and their careers in Information Technology industries.

References

[1] Hyoungmee Hahn, Jungpil Hahn, Jinwoo Kim, "A Cognitive Engineering Study on the Development of an Object-Oriented ProcessModeling ",Proceedings of the Thirtieth Annual Hawaii International Conference on System Science, 1997, (pp 199-209).

[2] Yen-Lin Chen, Chuan-Ming Liu, Chuan-Yen Chiang, Shyan-Ming Yuan and Jenq-Haur Wang, "Building Communication Software: a Project-based Approach for Teaching C++ Object-oriented Programming", International Journal of Innovative Computing, Information and Control, ISSN 1349-4198, November 20, 2012.

[3] Yen-Lin Chen, Chuan-Yen Chiang, Yo-Ping Huang, Shyan-Ming Yuan, "A Project-Based Curriculum for Teaching C++ Object-Oriented Programming", 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 4-7 Sept. 2012, (PP 667-672).

[4] Wellington, C., "Designing an ABET accreditable computer science bachelor's program within the constraints of a teaching university," *Frontiers in Education, 2004. FIE 2004. 34th Annual* , vol., no., pp.T3G/18,T3G/23 Vol. 1, 20-23 Oct. 2004

[5] Yulia,Rudy Adipranata,"Cooperative Learning Method Based On Game Design and Visual Object Oriented Environment to Teach Object Oriented Programming Course",Informatics Department Petra Christian UniversitySurabaya, Indonesia.ICETC.

[6] Yen-Lin Chen, Chuan-Ming Liu, Chuan-Yen Chiang, Shyan-Ming Yuan And Jexq-Haur Wang, "Building Communication Software: A Project-Based Approach For. Teaching C++ Object-Oriented Programming", International Journal Of Innovative Computing, Information And Control (IJICIC)2013, ISSN 1349.4198,pp 3415 3436

[7] Michael Kölling, "Using BlueJ to Introduce Programming", University of Kent,Canterbury,Kent, CT27NF,UnitedKingdom.

[8] David J. Barnes & Michael Kölling, "Objects First with Java", Fifth edition, Prentice Hall / Pearson Education, 2012, ISBN 978-013-249266-9.