

The Results of Classification of Lab Assignment in Object Oriented Programming and Database Management System Lab: A Case Study

Vani Yelamali
vani.yelamali@kletech.ac.in
KLE Technological University, Hubli.

Abstract: Object-oriented program, an emerging dominant paradigm in designing and constructing large and complex information systems. The requirement of industry expectations are not fully meet, which is evident through students learning. Therefore to reduce the gap between industry expectation and students understanding a deeper understanding of programming is essential and hence taught to undergraduate engineering students.

This paper talks about the classification of lab assignments, as analyzing the given software application problem, in six different phases. 1. Design phase 2. Object-oriented programming implementation 3. DBMS implementation 4. Integration of OOPS with DBMS 5. Structured enquiry 6. Open-ended project. This classification of assignment statements improves the involvement of students, which helps to meet the industry expectations.

Rubrics based assessment is followed. The result of batch 2019 in comparison with 2018 has shown marked improvement with this new approach. This results due to the above said approach. The feedback of student through questionnaire shows that they have acquired knowledge by improving their skills in terms of designing a class diagram, OOPS concepts, and integrating OOPS concepts with DBMS.

Key Words: Object-oriented programming, DBMS, UML, SDLC.

Introduction: The challenge here is to acquire programming knowledge (OOPS concepts) which needs various skills and knowledge. The students should acquire problem-solving skills along with programming skills. In OOPS, for a given statement of problem, the study comprises of identifying classes, attributes, and components of classes, objects

Sahana Beelagi
sahana.beelagi@kletech.ac.in
KLE Technological University, Hubli.

and methods. After study, the solution has to be represented using a standard diagrammatic

representation that is UML which enhances comprehension of OOPS concepts and it's easier to transport UML class diagram to JAVA/.net programming code. The aim of standard notation helps others to understand design solution and implementation. The database has been designed and implemented using the OOPS tools: conceptual model and logical model which involves the creation of ER-Models, designing database schema for a given scenario, and effectively manipulate the data stored in database and enhance database retrieval performance. The challenge involved here is to make students understand how to integrate OOPS concepts with DBMS concepts.

A. Classification of lab assessment :

No. of Lab Session	Experiments	Marks	Measuring parameter
02	Design of software application for an automation system	10	Analyze
03	Implementation of software application using the concepts of OOP	15	Implement
02	Implementation of a database to satisfy the requirement of an automation system	10	Implement
01	Design and implementation of	05	Analyze

	sensor, actuators and controls classes using the UML diagrams and concepts of OOP respectively		e
		05	Implement
01	Design and implementation database schema for sensors, actuators, and controls using the ER models and concepts of database, respectively	05	Analyze
		05	Implement
02	Implement an application that utilizes previously generated JAR/DLL files and database schema to store data from sensors and control the actuators	05	Analyze
		10	Implement
02	Implement an open-ended project using .NET/Java/python/DB concepts for an automation application	05	Analyze
		10	Implement
		05	Report

Table-1: Classification of lab assessment Rubrics are used to assess the students learning. The assessment matrix helps the evaluation process for uniform grading of the students.

Rubrics:

	Weightage	Description
Poor	00 % ~ 40 %	- Partially fulfil the requirements as mentioned under the deliverable heading - Fails to clarify the content of deliverable - Late to lab
Good	41 % ~ 80 %	- Fulfil most of the requirements as mentioned under the deliverable heading - Clarifies the content of deliverable that meets the expectation - Punctual to lab
Excellent	81 % ~ 100 %	Fulfil all the requirements as mentioned under the deliverable heading, including attempts to address the extension activities, if any

	- Clarifies the content of deliverable that exceeds the expectation - Punctual to lab
--	--

Table-2: Assessment rubrics

Implementation:

Acquiring knowledge of Object-oriented concepts is vital for engineering students which entitle to solve the complex business application. The 2nd year students of the Automation and Robotics Department are offered this lab course and this knowledge makes students learning easier in the coming semester to implement project at several stages like Course project, Mini project, Minor project, and a Capstone project. Keeping this in mind the lab course is structured with the following course outcomes.

Course outcomes:

1. Evaluate the given scenario, design, and model the system using UML.
2. Write a program in JAVA or .NET environment by using Object-oriented programming fundamentals like the composition of objects, operator overloading, inheritance, and polymorphism for a given scenario.
3. Apply programming knowledge, including scripting elements to develop real-world applications.
4. Identify different issues involved in the design and implementation of a database system.
5. Create data models using Enterprise Relationship(ER) notations.
6. Solve the problems related to data manipulation language to query, update, and manage a database.
7. Implement an open-ended project using Object-oriented programming and database concepts for automation and robotics application and submit a technical report.

To perceive the set outcome of the lab course we adopted a method of classification of lab assignments. The classification of lab assignment is as Demonstration, Exercises, Structured-Enquiry, and Open-Ended. Demonstration exercises form the base for the lab course activities, here the students get used to tools, use cases, creating class diagram and sequence diagram. The exercise kind of problem statement states the complete details of classes,

inheritance, constructors, method overloading/method overriding, and the output.

Example: Scenario for Painting Robot Application.

1. A truck with a container is parked on the parking platform next to the paint robot.
2. The operator identifies the container type.
3. The operator loads a suitable program (with G codes). This program corresponds to container type.
4. The operator manually operates the robot and sets the Zero position, which is the bottom right of the container in the front view.
5. The operator fills the paint bin
6. The operator starts the painting process
7. It takes roughly around 25 minutes to paint one side of the container.
8. The truck is driven once the painting is finished

As a part of demonstration exercises the students have to draw a class diagram for the scenario explained and translate the conceptual model to an object-oriented program. The outcome of this kind of problem statement is that the students learn to apply concepts of object-oriented to solve the problem.

Structured Enquiry:

These kind of problem statement activate student involvement more to build database, to identify and implement the classes, components of classes, and the relationship between the classes: is-a/has-a, method overloading/overriding from a given domain. Structured Enquiry has no predefined answer. The conclusion is solely based on the student's study. It helps students in classification and interpretation.

Example: Implement an application that utilizes previously generated JAR/DLL files and database schema to store data from sensors and control the actuators

Open-Ended:

This kind of assignment is assigned to a students in a team of 2. Here they have to design and implement an open-ended project for an automation application of their choice. The application should include GUI, java/.net code, and Database using Mysql workbench.

Table-3: shows the assessment rubric for open ended experiment.

Table- 3:Open-Ended Experiment Rubric

Sr.No	Parameter	Marks
1	Analysis and integration design	5M

2	Demonstration of application and Viva	10M
3	Report	5M

Result Analysis:

The conceptual model introduced for the batch 2019-20 shows a marked improvement as compared to 2018-19 batches, where they followed traditional method. The traditional method (2018-19) ensures surface learning which concentrates more on program syntax and semantic. The student's results are compared grade-wise. The Fig-1 depicts decrease in grades E, D and increase in grade C, this is because the proposed method ensures deeper understanding with the support of classification of lab assignments.

There is an immense improvement in the performance of the students even though there is no vast difference in the results. This has drastically improved in understanding and analyzing the problem and improving the coding standard. The real-time problem scenario made them analyze and understand the OOPS concepts clearly and to integrate OOPS concepts and DBMS concepts.

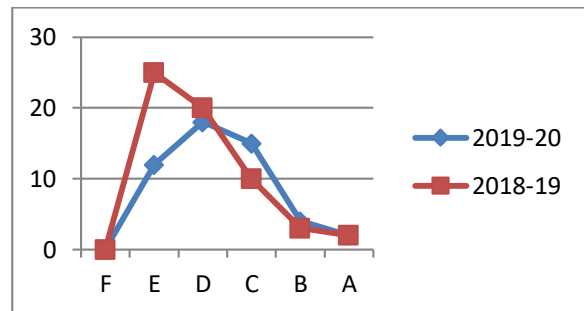


Fig-1: Comparison of students result

In Fig-1 it is observed that, there is increase in grades B and C and result remain same in grades A and F. This shows that the present model helped student to understand and analyze the problem.

A. Students feedback: Feedback form:

Sr.No	Questions
1.	The lab course expectation was made known well in advance.

2.	The syllabus, pedagogy, and examples to design the class diagram are satisfactory.
3.	The classification of the given real-time problem helped to learn the course content and apply the concepts learned.
4.	For a given problem drawing and designing a class diagram and ER-diagram helped to analyse and for writing code and creating DB.
5.	The integration of DB and code improved design analysis of real-time problems.

The feedback is taken for the present approach and is analyzed; the Fig-2 shows the graphical analysis of the feedback.

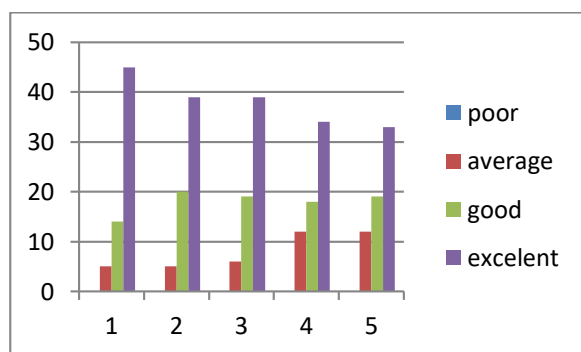


Fig-2: Feedback Analysis

Conclusion:

Learning of Object-oriented programming along with database has added additional skills among students like learning the basic OOPS concepts and database languages and in addition to this integration of OOPS and database, apply the same to develop an open-ended project. The students ability on problem solving skills, analyzing the given code, and performing proper documentation and indentation of the code and also integrate with the database was enhanced by following the pedagogical practices. Further, a target can be set to reach a benchmark of getting certification like OCAJP (Oracle Certified Associate Java Programmer).

References:

- [1]. K.M.M Rajashekharaiyah, Mahesh S. Patil & Dr. G.H Joshi "Impact of classification of lab assignments and problem-solving approach in Object-Oriented Programming Lab course: A case study" 2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE)
- [2]. Dr. Shivanand M. Handigund, Siddappa G. Makanur, Dr. M. Sreenivasa Rao "Integration of Object-Oriented Host Program with Network DBMS" The 2015 International Conference on Soft Computing and Software Engineering (SCSE 2015)
- [3]. Vidya Handur, Priyadarshini D Kalwad "Integrating Class and Laboratory with Hands-on Programming: Its Benefits and Challenges" 2016 IEEE 4th International Conference on MOOCs, Innovation, and Technology in Education
- [4]. Bennedsen, J., & Caspersen, M. E. (2004). Teaching Object-Oriented Programming- Towards teaching a Systematic programming process, 18th European Conference on Object-Oriented Programming (ECOOP). June 14-18, Oslo, Norway.
- [5]. Kölling, M. (1999). The Problem of teaching Object-Oriented Programming, Part 1: Languages. Journal of Object-Oriented Programming, 11(8):8-15.
- [6]. National Board of Accreditation New Delhi-110003: Manual for Accreditation of Undergraduate Engineering Programs, 1 edn (Mar. 2012).
- [7]. Jens Bennedsen. "Learning Object Orientation by professional adults", International Journal of Continuing Engineering Education and Life-Long Learning, 2006.