International Journal of Research

International Journal of Research (IJR)

e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 09, September 2015

Available at http://internationaljournalofresearch.org

Relevance of Object-Oriented Programming in Computer Education

Onu F.U.[1]; Onwubuya S.K.[2]; Oko F. J.[2] & **Igboji K.O.[1]

Department of Computer Science, Ebonyi State University – Abakaliki.

Department of Computer Education, Federal College of Education (T) – Asaba.

e-mail: otubok@yahoo.com

ABSTRACT

There is relatively sheer decline in programming culture leading to non-availability of high quality solution systems. But this can be engineered by intelligent utilization of Object-Oriented Programming (OOP) language concepts. This study unveiled ingenuity of oops as goodly facilitator of computer education, given its numerous channels with which to interface computer-aided designs, adaptable tutoring systems, e-learning and various e-transaction platforms across the globe. It is a high-level computer language that uses self-contained, modular instruction sets for defining and manipulating aspects of a computer program. Predefined instruction sets called objects are often used to define variables, data structures, and procedures for executing data operations. OOPs prominent concepts of inheritance, abstraction, encapsulation and polymorphism provide mechanism for solving complex problems. They make for modularity, extensibility and reusability as hallmark for improved software development strides. Interviews on experts, experience from routine practical uses and observations from some real-time application are sources of facts relied on. Economic impacts and viability of oop anchor on high quality products, improved software maintainability, low cost and faster development. Hence, a paradigm switch-over to this technique by many enterprising software developers.

KEYWORDS: programming; inheritance; modularity; encapsulation; polymorphism; class; object

INTRODUCTORY BACKGROUND

Object-oriented programming language could be defined as a user-friendly technique with robust mechanism for holistic tackle plan in solving complex problems. Sarah and Stacey (2000) said it is a programming method that combines data and instructions for processing the data into a self-sufficient object that can be used in other programs. Object-oriented programming emphasises object management and application design through mechanism such as classes and inheritance. Inherent potentials such as modularity, extensibility, and reusability are key factors to maximizing the ingenuity of OOP's. While modularity provides separation of duties in object-based program development, extensibility includes new attributes/behaviours of objects and reusability makes for object reuse, within and across applications.

There is relatively sheer decline in programming culture, leading to non-availability of high quality solution systems. But this can be engineered by intelligent utilization of oop concepts. This article became imperative as it portends a credible resort for new developers in matching the relevance of Object-Oriented Programming (OOP) in computer education. Succinctly, it unveils the ingenuity of oops as goodly facilitator of computer education. Economically, oop is adjudged to be impactful and viable in terms of high quality products, improved software maintainability, low cost and faster development. So, the educational sector of any society stands to be boosted where its concept are appropriately utilized in setting up interactive teaching and learning platforms.

Classical Description of OOP's

Object-oriented programming is a widely adopted programming technique that uses interactive objects to model and solve complex programming tasks. In an object-oriented driven system, the design is represented by objects, and objects have two sections thus: fields (instance variables) and methods. While Fields represent what an object is, Methods represent how an object is used. These fields and methods are closely tied to real world characteristics and using the object by means of its methods.

Sarah and Stacey (2000), itemized four classical facts that further describe OOP's (pronounced as "oops!"):

- OOP is a programming method that combines data instructions for processing data into a self-sufficient "object" that can be used in other programs (object is emphasized).
- An object is a book of preassembled programming code that is a self – contained module. The module contains, or encapsulates both chunk of data and the processing instructions that may be performed on that data.
- When an object's data is to be processed: sending the message: once the object becomes part of a program the processing instructions may or may not be activated. That happens only when a message is sent. A message is an alert sent to the object when an operation involving that object needs to be performed.
- How the object's data is processed (the methods): The message needs only identify the operation. Method refers to instructions about the operations to be performed on data within the object.

Types of Object-Oriented Programming Languages

There are barely two popular object-oriented programming languages which includes Java and C++. However, other well-known object-oriented programming languages that evolved are Objective C, Perl, Python, Javascript, Simula, Modula, Ada, Smalltalk, Delphi, C#, Ruby, PHP and the Common Lisp Object Standard. One of the main contributions of Smalltalk was that language constructs were combined with the flexibility of Lisps systems (Ole et al, 1993).

VIABLE CONCEPTS OF OBJECT-ORIENTED PROGRAMMING

Object-oriented programming parades viable concepts which go under the jaw-breaking names of encapsulation, inheritance, abstraction, and polymorphism. Actually, these terms are not as fearsome as they look.

- **Encapsulation**: it simply means that an object contains (encapsulates) both data and the instructions for processing it, as we have seen. Once an object has been created, it can be reused in other programs. An object's uses can also be extended through concepts of class and inheritance.
- **Inheritance**: Inheritance is the method of passing down traits of an object from classes to subclasses in the hierarchy. Once an object is created, you can use it as the foundation for similar objects that have the same behaviour and characteristics (it shares behaviour). All objects that are derived from or related to one another are said to form a class, each class contains specific instructions (methods) that are unique to that group. Class can be arranged in hierarchies of classes and subclasses. Thus, new objects can be created by inheriting from existing classes.
- **Abstraction**: is a concept that focuses on idea, quality and properties but hides certain irrelevant details. It reduces complexity by using single construct to replace multiple entities that does similar thing.
- **Polymorphism**: Polymorphism allows a programmer to create procedures for objects whose exact type is not known in advance but will be at the time the program is

actually run on the computer. It simply means "many shapes" (i.e. many operations can be performed using wide range of different types of things). In oop, polymorphism means that a message [generalised request] produces different results depending on the object that it is sent to.

Economic Importance/Relevance of Object-Oriented Programming

It is pertinent to note that oop is a fundamental modelling framework in computing – thus providing a critical path to greater software productivity. Some definite benefits of object-oriented programming that permeates various sectors of developing economy include:

- Improved software-development productivity: Object-Oriented Programming is modular, as it provides separation of duties in object-based program development. It is also extensible, as objects can be extended to include new attributes and behaviours. Objects can also be reused within and across applications. These factors–modularity, extensibility and reusability made oop improve software-development productivity over traditional procedure-based programming techniques.
- Improved software maintainability: Due to the above reasons, object oriented software is also easier to maintain. Since the design is modular, part of the system can be updated in case of issues without a need to make large-scale changes.
- Faster development: Reuse enables faster development. Object-oriented programming languages come with rich libraries of objects, and code developed during projects is also reusable in future projects.
- Lower cost of development: The reuse of software also lowers the cost of development. Typically, more effort is put into the object-oriented analysis and design, which lowers the overall cost of development.
- Higher-quality software: Faster development of software and lower cost of development allows more time and resources to be used in the verification of the software. Although quality is dependent upon the experience of the teams, object oriented programming tends to result in higher-quality software.

However, a few limitations have been observed in terms of steep learning curve, where the thought process involved in oop may not be natural for some people, as a result takes time to get used to. Also, there is relative complexity in creating programs based on interaction of objects. Some of the key programming concepts such as inheritance and polymorphism can be challenging to comprehend initially. And lastly, oop typically involves larger program size (more lines of code), than procedural programs.

STRATEGIC REVIEW

The earliest programming languages preceded the invention of the digital computer and were used to direct the behaviour of machines such as Jacquard Looms and Player Pianos. The purpose of which is to find a sequence of instructions that will automate performing of specific task or solving a given problem. According to Kendler and Krivy (2011), programming language is a formal constructed language designed to communicate instructions to a machine, particularly a computer. Programming languages can be used to create programs to control the behaviour of a machine or express algorithms. The process of programming thus, often requires expertise in many different subjects including knowledge of the application domain, specialized algorithms and formal logic.

According to Bjarne (1991), object-oriented programming is programming using inheritance, and data abstraction is programming using user-defined types. Most object-oriented programming languages are based on the class/subclass mechanisms (inheritance) which first appear in Simula.

Object-Oriented Programming Vs Other Programming Techniques

The use of OOP's has tremendously improved software development productivity over traditional procedure-based programming techniques. This is largely due to the availability of these three features of modularity, extensibility, and reusability. Until recently, programs were thought of as a series of procedures that acted upon data. A procedure or function is a set of specific instructions executed one after the other. The data was quite separate from the procedures and the trick in programming was to keep track of which functions called which other functions, and what data was created.

On the other hand, structured programming refers to the technique that produces programs with clean flow, clear design, and a degree of modularity or hierarchical structure. It has prominent benefits such as ease of maintenance and ease of readability by other programmers. Though, structured programming remains an enormously successful approach for dealing with complex problems, by the late 1980s, some of the deficiencies of structured programming had become too glaring. Firstly, it is natural to think of your data and what you can do with your data as related ideas. Secondly, programmers found themselves constantly reinventing new solutions to old problems. This is called "re-inventing the wheel" which is the opposite of re-usability.

Obviously, Object Oriented Programming is a programming paradigm based on the concept of "objects" which are data in the form of fields, often known as attributes and code in the form of procedures often known as methods, attributes and code. A distinguishing feature of objects is that an object procedure can access and often modify the data fields of the objects with which they are associated. According to Kindler and Krivy (2011), there is significant diversity in object-oriented programming, but most popular languages are class-based, meaning that objects are instances of classes, which typically also determines their type.

Many of the most widely used programming languages are multi-paradigm programming language that support object-oriented programming to a greater or lesser degree, typically in combination with imperative, procedural programming. Mitchell (2013), states that Object-Oriented Programming language typically shares low-level features with high-level procedural programming languages. Interestingly too, languages that support object-oriented programming typically engaged inheritance for code reuse in the form of either classes or prototypes. Those that use classes support two main concepts of Objects and Classes. According to Booch (1997), each object is said to be an instance of a particular class. For example, an object with its name field set to "Mary" might be an instance of class employee and lots more.

PROGRAMMING AND COMPUTER EDUCATION

Programming: In the simplest terms, computer programming is the act/art of writing program. And program refers to logical set of instructions that directs a computer to perform some processing function or combination of functions. Shaun (2014), says programming involves activities such as analysis, developing understanding, generating algorithms and verification of requirements of algorithms including their correctness and resource consumption and implementation. It leads to original formulation of computing problem and generating sequence of code in precise order which will be executed to produce a target result.

Indeed, programming is and has been the underlying technology that drives computer education in this century. Barely all discipline in the educational system of both developed and developing economy thrives on computing. It is instructive to state that the proliferation of computing infrastructure to other educational discipline was achieved through programming techniques.

More importantly, the manoeuvre and penetration of computer into virtually all facet of human endeavour is a clear function of programming interface. Standard computer application software packages are products of programming expertise, and they created phenomenal doorway into various educational discipline and by extension other strata of life.

Impacts of OOP's in Computer Education

The revolution of oop technique in computer education speaks loudly across disciplinary divide. Educational system is meant to create robust environment for learning and research innovations and ways to transform same to societal gains. It therefore becomes imperative to explore viable option afforded by the ingenuity of the OOP's in generating real-time resource base that can beat down unforeseen challenges.

Computer education has gone far beyond classroom or laboratory interfaces to a more technological drive via multimedia platforms orchestrated by oop mechanisms. In the past, computer education was limited to those students with exceptional mathematics aptitude. Unfortunately, in many schools computer education has become an educational ghetto (Alfred Thompson, 2007). Notwithstanding, great feats have been accomplished using computer in education. At the front burner of these accomplishments is a veritable tool called object-oriented programming – seen as today's "magic maker" device.

The ingenuity of oops as goodly facilitator of computer education created numerous channels with which to interface computer-aided designs, adaptable tutoring systems, e-learning and various e-transaction platforms across the globe. Computer education improves student's research skills and encourages them to explore available means to intensify information search via the Internet. Suffice it to say that incorporating computer education in schools has inspired students to undertake careers in technology and enhance their understanding of how computer technology impacts peoples' daily lives. Computer-based class allow students to put their creativity to use. Most of such creative instincts are aroused by intense study of programming technique of which oop is foremost. For example, a class work can involve mini-project to develop typical solution systems which obviously demands programming knowledge, which subjects them to think virally to produce desired results.

DISCUSSION AND EVALUATION

As programming in the large technique, oop focuses on components and composition mechanism to establish imperative computing within objects and its behaviour. Practicable and literary evidence as opined and outlined by experts in the field of oop clearly showed that, the technique has tremendously impacted on the fabrics of computer educational system around the globe. It has become the stream hub for development in the education sector. A technique that ostentatiously treats data and the procedures that acts upon the data as a single object – a self contained entity with an identity and certain characteristics of its own.

The popularity of OOP will continue to increase as the backlog of yet unwritten program rise in large business and software houses. Today, applications are becoming larger and more complex. As size and complexity of project increases, so does the number of experts and years required to complete it, which unduly overstretches a project. Sarah and Stacey (2000), says that one of the few proven methodologies for handling this increase in project complexity is object-oriented programming. Indeed, oop as an outcome of many years of evolutionary process, that scaled several competency test has become and will continue to be the buzzword in software development productivity. There is however the likelihood of concurrency in software engineering adventure involving flexibility and efficiency as well as structure and modularity. So, one could not readily project a limit to its future advances.

SUMMARY AND CONCLUSION

Object Oriented Programming is centred on creating objects rather than procedures and functions. It saves development time and cost by reusing code. Once object class is created, it can be used in other applications having the same functionality with it. Programs written in Object Oriented Programming language is easier to debug due to the fact that reused objects have already been tested and also classes can be tested independently.

Due to the fact that Object Oriented Programming enables one to remain close to the conceptual, higher-level model of the real-world problem you are trying to solve, you can manage the complexity better than with approaches that force you to map the problem to lit the features of the language. You can take advantage of the modularity of objects and implement the program in relatively independent units that are easier to maintain and extend. Again, code sharing among objects through inheritance is an added advantage (Oyefola et al, 2014).

REFERENCES

[1.] Alfred, T. (2007). Computer Science Teacher. USA: Blogs Publication Limited.

[2.] Bjarne, S. (1991). What is “Beyond Objected-Oriented Programming”?. New Jersey: Murray Hill.

[3.] Booch, G. (1997). Object-Oriented Analysis and Design with Applications. USA: Addison-Wesley. 0-8053-5340-2.

[4.] Clemens, S. (2013). Beyond Objected-Oriented Programming. USA: Addison – Wesley Longman Publishing Co. Inc. ISBN 02017 45720.

[5.] Elizabeth, S. (2013). Kids Need Computer Science Education. USA: Love To Know Corp.John, M. (2013). Concepts in Programming Language. UK: Cambridge Kindler, University Press. ISBN 0-521-78098-5, P.278. List: Dynamic Dispatch, Abstraction, Subtype Polymorphism and Inheritance.

[6.] Kindler E. and Krivy, I. (2011). Object-Oriented Simulation of Systems with Sophisticated Control. USA: International Journal of General Systems. PP. 33-343.

[7.] Lewis, J. and Loftus, W. (2008). Java Software Solutions Foundations of Pierce, Programming Design. 6th edition. USA: Pearson Education Inc. ISBN 0--532 05-8, Section 1.6 “ObjectOriented Programming”.

[8.] Ole L. M., Birger M.P. and Kristen N. (1993). Object Oriented Programming in the Beta Programming Language. Oslo: Norwegian Computing Centre

[9.] Oyefola G., Oki M. and Balogun A.M. (2014). Object Oriented Technology. Lagos: National Open University of Nigeria

[10.] Pierce, C. (2002). Type and Programming Language. USA: MIT Press. ISBN O-262-16209-1, Section 18.1

[11.] Sarah E.H. and Stacey C.S. (2000). Computers, Communications and Information (Comprehensive Version). USA: McGraw-Hill Higher Education. Pgs. 10.24-10.25

[12.] Shaun, B. (2014). What is Programming. UK: Wiley Corporation Limited.