ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 1: Programming		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Duong Hong Duc	**Student ID**	BH01425
Class	SE07101	**Assessor name**	Nguyen Thanh Trieu

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student's signature	Duc

Grading grid

P3	P4	P5	P6	M2	M3	M4	D2	D3	D4

☼ Summative Feedback:	☼ Resubmission Feedback:	
Grade:	Assessor Signature:	Date:

Lecturer Signature:

Table of Contents

I.Introduction

Now the research and development team you work with have been tasked with further investigation into how best to build more efficient, secure software for calculate water bill. You have been asked to look into programming paradigms and the advantages and disadvantages of using different programming language approaches.

You will need to create a report covering findings from research into the characteristics of different programming paradigms – procedural, object-orientated and event-driven programming. After that, you and your team are given a more challenging task to create a fully working, secure application that has been developed using an IDE and adheres to coding standards for a detailed business problem.

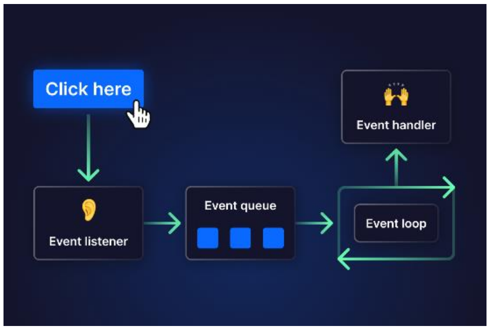
II.Body

1. definition of Event Driven, Object Oriented and Procedural

- Event driven:

Event-driven programming is a programming paradigm in which program execution is determined by new user events (mouse clicks, keypresses), sensor outputs, or message passing from other programs. Programmers use Event-driven programming in graphical user interfaces and other applications that focus on performing user actions in response to user input (user clicks).

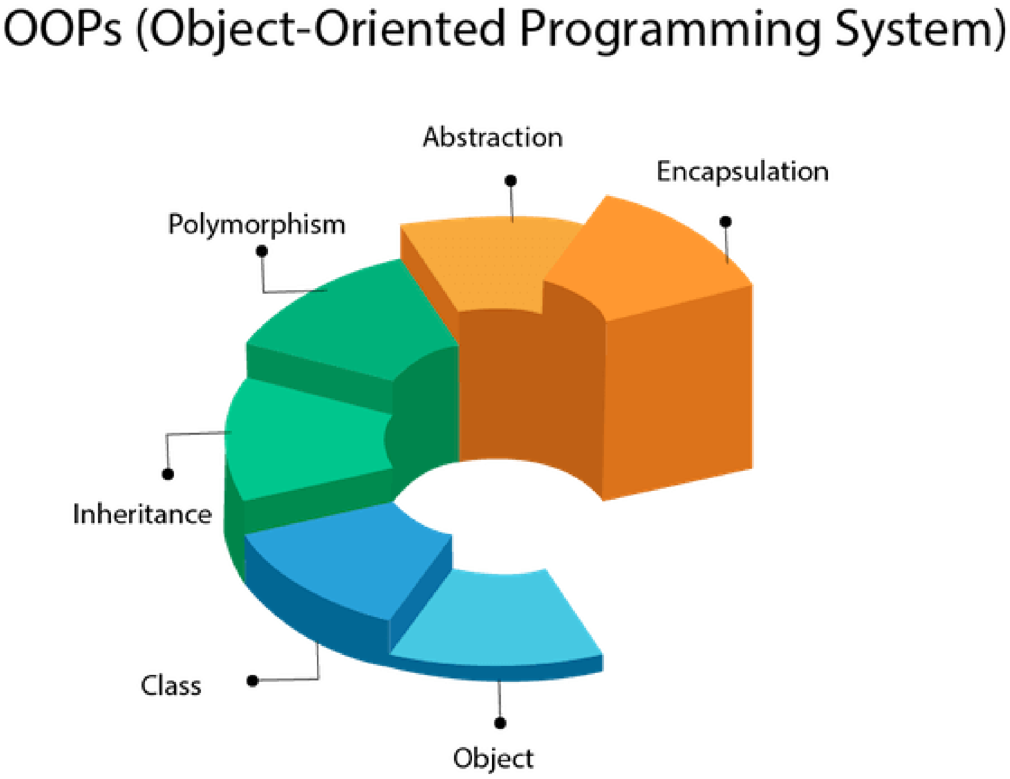
We can also define event-driven programming as a way of building a computer program, in which, in the essential function of the program, the main event loop of the application is explicitly highlighted in the code, the body of which consists of two parts: fetching the event and event handling. In an event-driven application, the main loop listens for events and triggers a callback function when one of those events is detected.



- Object oriented

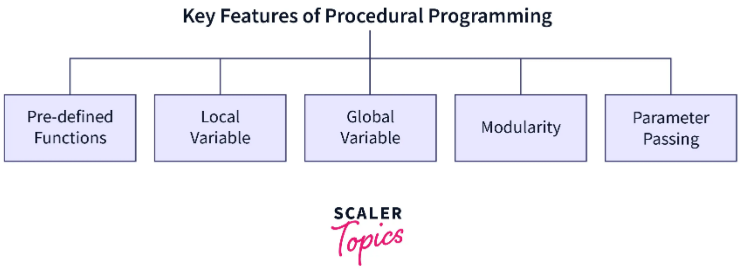
Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

OOP focuses on the objects that developers want to manipulate rather than the logic required to manipulate them. This approach to programming is well-suited for programs that are large, complex and actively updated or maintained. This includes programs for manufacturing and design, as well as mobile applications; for example, OOP can be used for manufacturing system simulation software.



- procedural

Procedural Programming can be defined as a programming model which is derived from structured programming, based upon the concept of calling procedure. Procedures, also known as routines, subroutines or functions, simply consist of a series of computational steps to be carried out. During a program's execution, any given procedure might be called at any point, including by other procedures or itself.



2. characteristics and features

- event driven

Service-Oriented

Service-focused is a key feature in event-driven programming used to create programs created for services, and it does not slow down the computer. The service-oriented only take in little of the computer control vitality, and usually, services run in the backdrop of OS.

Time-Driven

In motivated event programming, time influenced is a paradigm. From the code that runs on a period trigger, time motivated can be considered a specific code that works in a particular time, which could be once one hour, once a week, or once a month. This means from the pre-set to do the activity. For example, house windows revise is the exemplary case of time motivated, which individual can establish when to revise or when to check and download the revise.

Trigger Functions

Trigger functions in event-driven coding are functions that decide on what code to perform whenever there is a specific event occurs. Programmers use these functions to select which event handler to use for the function when you can find specific event occurred.

Events

Events include mouse, computer keyboard, and user interface, which events have to be triggered in this program. This means the user has to interact with an object in this program, for example, click a button by a mouse, use the computer keyboard to choose a button, etc.

The Simplicity of Programming and Ease of Development

Event-driven programming is simple and much more straightforward to program than another kind of programming as it's very aesthetic. For example, you can place a button by selecting it and place it onto a form and write a code for this. Event-driven development is also easy for the user to place a pre-written code script into an existing application. It allows the user to pause the code while running. Which makes expanding using event-driven programming is easy.

- Object oriented

Enables code reusability

The idea of inheritance is one of the critical concepts offered by object-oriented programming. A class’s attributes can be passed down through inheritance, eliminating the need for duplication of effort. Doing this prevents the problems associated with repeatedly writing the same code.

Thanks to introducing the idea of classes, the code section can be used as many times as necessary in the program. A child class that uses the inheritance method inherits the parent class’s fields and methods. One can readily alter the parent class’s available methods and values.

Increases productivity in software development

We can create programs from pre-written, interconnected modules rather than having to start from scratch, which would save time and increase productivity. Thanks to the OOP language, we can break the software into manageable, discrete problems. Because it allows for the division of labor in the creation of object-based programs, object-oriented programming is modular.

It is also extendable, as you may add new characteristics and actions to objects. One can utilize objects in several applications. Object-oriented programming increases software development productivity, compared to conventional procedure-based programming techniques, due to modularity, extensibility, and reusability.

Makes troubleshooting simpler

When object-oriented programming is used, troubleshooting is made simpler since the user knows where to look in the code to find the source of the problem. Since the error will indicate where the issue is, there is no need to inspect additional code areas. All objects in object-oriented programming (OOP) are self-constrained, which is one benefit of employing encapsulation. DevOps engineers and developers gain a lot of advantages from this multimodal behavior because they may now work on several projects at once with the benefit of avoiding code duplication.

Reinforces security

To maintain application security and provide vital data for viewing, we are filtering out limited data through data hiding and abstraction mechanisms. The concept of data abstraction in OOPS allows only a small amount of data to be displayed to the user, which is one of OOP's strong points.

When only the necessary info is accessible, the rest is not. As a result, it makes security maintenance possible. Another set of OOP's advantages in Java's idea of abstraction is used to conceal complexity from other users and display the element's information per the requirements.

- procedural

Predefined functions:

A predefined function is typically an instruction identified by a name. Usually, the predefined functions are built into higher-level programming languages, but they are derived from the library or the registry, rather than the program. One example of a pre-defined function is 'charAt()', which searches for a character position in a string.

Local Variable:

A local variable is a variable that is declared in the main structure of a method and is limited to the local scope it is given. The local variable can only be used in the method it is defined in, and if it were to be used outside the defined method, the code will cease to work.

Global Variable:

A global variable is a variable which is declared outside every other function defined in the code. Due to this, global variables can be used in all functions, unlike a local variable.

Modularity:

Modularity is when two dissimilar systems have two different tasks at hand but are grouped together to conclude a larger task first. Every group of systems then would have its own tasks finished one after the other until all tasks are complete.

3.comparison

Procedural and Event Driven describe the general workflow of the application or decision making logic while Object Oriented describes more the structure of the decision making logic:

- Procedural describes a sequential workflow of logic, in general there are many steps that must be performed in a sequence, there may be criteria between each step that might be dependent on the outcome from previous steps, however the sequence of logic is pre-determined and hard coded into the application.

- Event Driven describes a workflow where the system raises event messages, or responds to events raised from other systems. The application logic is executed in direct response to these events, in explicit contrast to Procedural Programming the timing of the events is considered not controllable and due to this many events may need to be serviced concurrently, this is in direct contract the procedural programming where each step needs to run to completion to be ready for the next step in the chain can be executed.

- Object Oriented Programming describes a style where the state of the system is modelled using classes that describe a set of metadata and the behaviours and interactions with other objects. We can create Instances of a class to create objects. In this way you can think of OO as first defining a series of templates, and then creating objects from those templates.

4.evaluation

Below is how the source code implements in all three paradigms:

- Generate events: Producer will need to create events and send them to a specific message queue.

- Subscribe event: Consumers will subscribe to message queues. Depending on the queue configuration, a message can be sent to multiple queues or just one queue. This results in the same message being received by two consumers.

- Retention configuration: Some queues will immediately delete a message once it is delivered to subscribers. Other queues may keep events in them. With these queues, what we need to do is configure how long messages are retained. We can configure the message lifetime, the number of messages in the queue or the total number of messages of all queue events.

III.Conclusion

IV.Reference

Sagar Bhatia (2022). What is Procedural Programming? Key Features of Procedural Programming. [online] Hackr.io. Available at: https://hackr.io/blog/procedural-programming.

BasuMallick, C. (2022). What is OOP (object oriented programming)? Meaning, concepts, and benefits. [online] Spiceworks. Available at: https://www.spiceworks.com/tech/devops/articles/object-oriented-programming/.

Kolesnikova, T. (2019). Event-Driven Development Features. [online] StudyBay. Available at: https://studybay.com/blog/event-driven-development-features/.