



PROGRAM TITLE: PROGRAMMING

UNIT TITLE: PROGRAMMING

ASSIGNMENT NUMBER: 1

ASSIGNMENT NAME: PROGRAMMING

SUBMISSION DATE: 04/12/ 2021

DATE RECEIVED: 02/11/2021

TUTORIAL LECTURER: NGUYEN VAN HUY

WORD COUNT: 2012

STUDENT NAME: LO VAN HIEN

STUDENT ID: BKC12444

MOBILE NUMBER: 0326660728

|  |
| --- |
| **Summative Feedback:**    **Internal verification:** |

Table of Contents

[**I. Procedural-oriented programming ( POP) 4**](#_Toc89462253)

[**1. What is Procedure Oriented Programming? 4**](#_Toc89462254)

[**2. Characteristics and properties of Procedure Oriented Programming 4**](#_Toc89462255)

[**3. Procedural programming example code: 4**](#_Toc89462256)

[**4. Advantages and Disadvantages of Procedural-oriented programming 5**](#_Toc89462257)

[**II. Object-orientated programming (OOP) 5**](#_Toc89462258)

[**1. What is Object-orientated programming? 5**](#_Toc89462259)

[**2. For example, object orientation: 6**](#_Toc89462260)

[**3. Object Oriented example code: 6**](#_Toc89462261)

[**4. Characteristics of object-oriented programming language 6**](#_Toc89462262)

[**5. Advantages and Disadvantages of Object-oriented programming 8**](#_Toc89462263)

[**III. Event-driven programming 8**](#_Toc89462264)

[**1. What is Event-driven programming? 8**](#_Toc89462265)

[**2. Application building steps in a directional manner event 9**](#_Toc89462266)

[**3. Event-driven programming example code: 9**](#_Toc89462267)

[**4. Characteristics of oriented programming language 9**](#_Toc89462268)

[**5. Advantages and disadvantages of event-oriented programming? 9**](#_Toc89462269)

[**IV. The relationship between them 10**](#_Toc89462270)

***LO2 Explain the characteristics of procedural, object- orientated and event-driven programming.***

# Procedural-oriented programming ( POP)

## What is Procedure Oriented Programming?

* + Procedure Oriented Programming is a traditional programming technique in which a program is divided into functions (subroutines).
  + Each subroutine can also be divided into many other subroutines to simplify their work.
  + For example, a program to input and display user information will be divided into two subroutines, an input and output program, if the input is complicated, the information input program can be divided into many different subroutines.
  + Procedure Oriented Programming languages: Pascal, C

## Characteristics and properties of Procedure Oriented Programming

* Program = Data Structure + Algorithm
* Data structure: how to describe, organize data.
* Algorithm: algorithm, steps to solve the problem.
* To link functions together, we often use global variables or pointers.
* The basic properties of structured programming are:
  + - * Focus on the work to be done (algorithms).
      * A large program is divided into subroutines, each of subroutines can be called one or more times in any order.
      * Most functions use common data.
      * Data in the system is transferred from one function to another.
      * Function that converts data from one form to another.
      * Using a top-down approach in program design.

## Procedural programming example code:

* A simple C hello world program

#include <stdio.h>

int main()

{

    printf("Hello, World!");

    return 0;

}

## Advantages and Disadvantages of Procedural-oriented programming

* Advantages
* Simple, easy to understand, Clear algorithmical thinking.
* The procedural programming languages are relatively much easier to learn as first programming language for the beginners.
* The straight forward program organization makes it ideal choice as a general purpose language
* The procedural programming language such as C Language is still being used for many application.
* The C language has extensive library of functions suitable for various applications. This prewritten code already used and tested is readily available to the programmer.
* The use of standard library functions bring down significant reduction in the overall development cost and time.
* The concept of pointers in the procedural programming C language allows low level memory operations.
* Disadvantages
  + - * The procedural programming is not suitable for large and complex software project.
      * It is difficult to represent the real world objects realistically in the procedural programming. Whereas , in object oriented programming , it is much easier to represent the real world objects.
      * It is difficult to protect the data from inadvertent changes since most data is generally global leading to the problem of spaghetti code.
      * The software maintenance is relatively difficult for a procedural programming software.
      * For procedural programming paradigm , the function is the most important component of the program and the data does not get the due attention.

# Object-orientated programming (OOP)

## What is Object-orientated programming?

* Object-oriented programming is a programming technique in which the program is divided into small parts called objects.
* The object in object-oriented programming will correspond to the entities in the problem, it will have properties, methods, Objects that can interact with each other.
* Object-oriented programming languages: Java, C#, Python, Ruby, Swift
* The basic properties of object-oriented programming: abstraction, packaging, polymorphism, inheritance.

## For example, object orientation:

* For Pac-Man, there are three objects: Pac-Man, a ghost, and a pac-dot.
* The ghost has states of:
  + colour
  + name
  + state (eatable or not)
  + direction
  + speed
* and behaviours of:
  + moving
  + changing state

## Object Oriented example code:

// The program is written in java

abstract class Animal {

    public abstract void Move ();

}

class Dog extends Animal {

    public void Move () {

        System.out.println("Dog running...........!");

    }

}

class Main {

    public static void main (String[] args) {

        Dog dog = new Dog();

        dog.Move();

}

}

## Characteristics of object-oriented programming language

* Encapsulation
* Packaging allows for concealment of information and the internal processing properties of the subject. Other objects cannot directly impact the data within and change the status of the object but are required through the public methods provided by that object
* This nature increases the security of the subject and avoids unintended data damage.
* Inheritance
* This is a very much used nature. Inheritance allows the construction of a new class (Subclass), inherit and reuse properties and methods based on the old class (superclass) already available
* The Subclass inherits the entire composition of the Father class and there is no need to define it. Subclasses can expand the inherited components or add new ones.
* Example:
  + Superclass is smartphone , has the attributes: color, memory, operating system
  + Subclasses are iPhone, Samsung, Oppo also have attributes: color, memory, operating system ...
* Polymorphism
* Polymorphism in OOP programming allows different objects to perform the same functionality in different ways.
* Polymorphisms in object-oriented languages make us more likely to reuse writing code and can change behavior flexibly depending on the topic.
* Polymorphism is mainly divided into two categories is: Runtime Polymorphism and Compile Time Polymorphism:
* Multimorphism of running time is the process of calling the method that has been overwritten during the implementation of the program. In the process, an overwritten method is called through the reference variable of superclass.
* Compile Time Polymorphism is using method overloading Since the same name is used for many methods, we must tell java which methods to call to do so, java relies on differences in the number of objects as well as the type of data of these objects to distinguish those methods of the same name.
* Example:
* Smartphone class, each model inherits the components of the parent class but the iPhone runs on the iOS operating system, while Samsung runs on the Android operating system.
* Abstraction
* Abstraction in OOP is a nature that does not express specifics but only names the problem. It's a process of hiding inner activities and showing only the essential features of the audience to the user.
* As such, abstraction is to hide information made from users, they only know the feature provided.
* Its main goal is to reduce complexity by hiding details that are not directly related to the user (the user here is not the end user but the programmer)
* Advantages of using abstraction for programming:
  + Abstraction allows programmers to eliminate the complex properties of the object by giving only the necessary attributes and methods of the object in programming, improving the maintenance of the system.
  + Abstraction helps us focus on the essential cores of the object instead of caring about how it performs.
* Abstraction offers many expanded features when used in combination witth polymorphism and inheritance in object-oriented programming.

## Advantages and Disadvantages of Object-oriented programming

* Advantages
* Data is not freely altered in the program as Procedural-oriented programming.
* When changing the data structure of an object, there is no need to change the source code of other objects, but just change some component functions.
* The source code can be reused through inheritance.
* Suitable for complex, large software.
* Rated as easy to learn, productive, simple, easy to maintain, easy to expand ...
* Highly secure, protects information through packaging.
* Disadvantages
* Difficult to reach when you're just starting to learn.
* Not suitable for solving simple problems.
* The program is slower and larger in size.

# Event-driven programming

## What is Event-driven programming?

* Event-driven programming is a programming model in which the flow of a program is determined by events such as user actions (clicks, keystrokes), sensor output, or messages from programs or streams
* Event-oriented programming is a method of building a system based on development. graphical interaction interface across objects and definition of system behaviors Through the events of the objects.
* Event-driven programming is the dominant model used in the graphical user interface and other applications (e.g., JavaScript web application) that focuses on taking certain actions to meet the user's input. Event-Oriented Programming Languages : java, c, c++, c# , php , python…..

## Application building steps in a directional manner event

* Create a form object on the app.
* Design user interaction interface by bringing objects into form.
* Select events that need to interact with users (Click, MouseOver,...).
* Write code for events.

## Event-driven programming example code:

//The example below is written in jQuery (Javascript).

$('#button').click(function(){

    alert('Message box [freetuts.net]');

});

// This code will start the event when clicking on the HTML card with the id as the button, it will show up a message box with the message box "Message box [freetuts.net]".

## Characteristics of oriented programming language

* Service Oriented
* Time Driven
* Event Handlers
* Trigger Functions
* Events
* Simplicity of programming and ease of development

## Advantages and disadvantages of event-oriented programming?

* Advantages
* One of the big benefits of event-driven programming is that it's very intuitive, flexible and naturally well-suited to applications whose control flow are based, not on its structure, but rather on internal or external events
* An event-driven application is a computer program that is written to respond to actions generated by the user or the system. In a computing context, an event is any identifiable occurrence that has significance for system hardware or software.
* Improve customer feedback experience.
* Disadvantages
* Since it brings the most value to GUI applications, some other use cases may not benefit from it.
* Some critics also say that event-oriented programming is complex to master and it's difficult when your application is simple and small.

# The relationship between them

|  |  |  |  |
| --- | --- | --- | --- |
| BASIS FOR COMPARISON | POP | OOP | EVENT-DRIVEN |
| Define | - is a traditional programming technique in which a program is divided into functions (subroutines). | - is a programming technique in which the program is divided into small parts called objects. | - is a programming model in which the flow of a program is determined by events such as user actions (clicks, keystrokes), sensor output, or messages from programs or streams |
| Interested | Function | Object | Event |
| Follows | top down approach. | bottom up approach. | approach ratherc |
| Access specifier | There is no access specifier. | access specifiers like private, public, protected.. | access specifiers like main loop, event handlers, asynchronous, processes, etc |
| Proper | less secure. | more secure | less secure. |
| Overloading | not possible. | Overloading is possible in . | Overloading is possible. |
| Based | unreal world. | real world. | real world. |
| Advantages | - Simple, easy to understand, Clear algorithmical thinking,  - easier to reach when you're just starting to learn. | - Easily manage code when there are program changes.  - The source code can be reused through inheritance. | - it's very intuitive, flexible and naturally  - Suitability for Graphical Interfaces  - Improve customer feedback experience. |
| Disadvantages | - If it is a large program, it is difficult to manage the code and Difficult to find Error | - Difficult to reach when you are just starting to learn.  - The program is slower and larger in size. | - Difficult to find Error  - Less Logical and Obvious  - high cost to develop application |
| Commonly used languages. | Pascal , C | Java, C# , Python, Ruby, Swift.. | Java , c , c++ , c# , php , python… |