Object Oriented Programming

#1 Introduction

Hilmy A. Tawakal &
Azhar Rasyad,Auzan Assidqi, Aditya Fitriadi

September 11, 2019

Table of contents

Objectives

Mata kuliah ini dimaksudkan untuk memberi kemampuan pada mahasiswa untuk dapat **memahami** *konsep* **pemrograman berorientasi objek**. *Materi perkuliahan mempelajari pseudocode dan konsep program pada bahasa pemrograman berorientasi objek yang meliputi: struktur program, variabel dan tipe data, struktur logika dan pengulangan, array, method static, object dan class, exception, collection class, generalisasi, inheritance, interfaces, polymorphisme, encapsulation, annotations, IO stream, pemrograman thread dan socket.*

Syllabus

1. Introduction
2. Program structure, Variable, Operator, data type
3. Logic control, Loop
4. Array and Static Method
5. Object and Class
6. Essential Class
7. Exception Handling
8. Mid Test
9. Inheritence & Abstract Class
10. Interface, Polymorphisme, Encapsulation
11. Collection Class
12. Generics and Annotations
13. File Stream
14. Thread
15. Java Socket Programming
16. Final Test

Prerequisite & References

Prerequisite:

- DDP
- SDA

References:

- Patrick Naughton, Java Handbook : Konsep dasar pemrograman java, McGraw- Hill/Osborne
- Ariesto Hadi Sutopo & Fajar Masya, Pemrograman Berorientasi Objek dengan Java, Graha Ilmu, 2005.
- Benny Hermawan, Menguasai Java 2 & Object Oriented Programming, Andi Offset, 2004.
- Ariesto Hadi Sutopo, Analisis Dan Desain Berorientasi Objek, J & J Learning, 2002.
- Isak Rickyanto, ST, Dasar Pemrograman Berorientasi Objek dengan Java 2 (JDK 1.4),Andi Offset, 2005.

How to get 'A'

Attitude:

- Active
- Do your assignments
- Cheating & Plagiarism = 'E'
- Be nice to Lab assistance

Proportion:

- 50% Assignment
- 20% Mid Test
- 25% Final Test
- 5% Attendance

How to get 'E'

Cheating in the context of this course is generally, but not limited to:

- Sharing and copying of code from other students or the Internet without mention the source.
- Any code making up your solution should be written and understood by you.
- Small quantities of template code will at times be provided by the instructor. You can use this code in submissions but should still be able to fully explain the function of all template code you use.
- Refer to but do not copy code from the examples given in class

What is OOP

- Object-oriented programming (OOP) is a programming language model organized around **objects** rather than "actions" and **data** rather than logic.
- Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data.
- Object-oriented programming takes the view that what we really care about are the **objects we want to manipulate** rather than the **logic required to manipulate them**.

Evolution

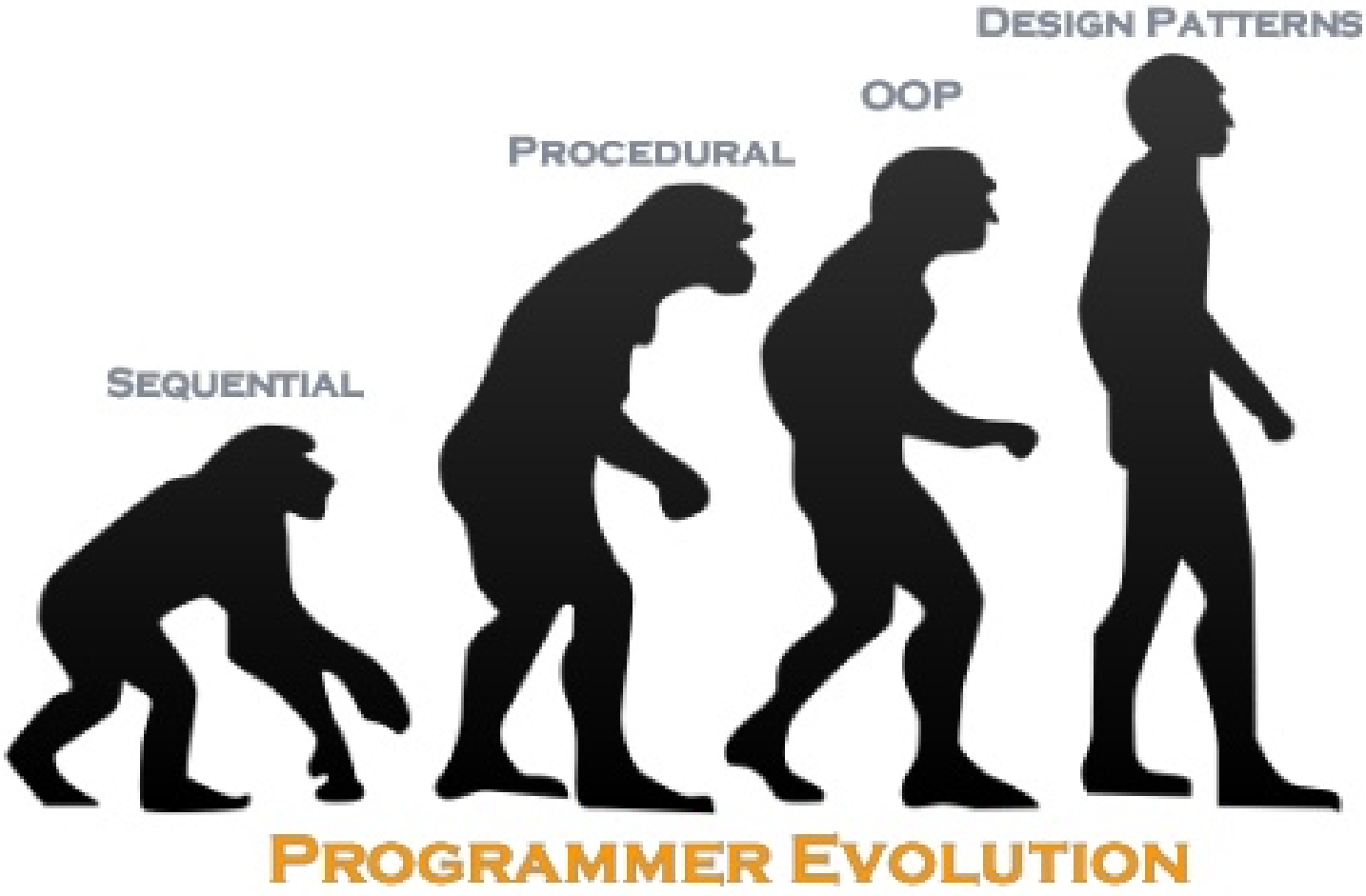
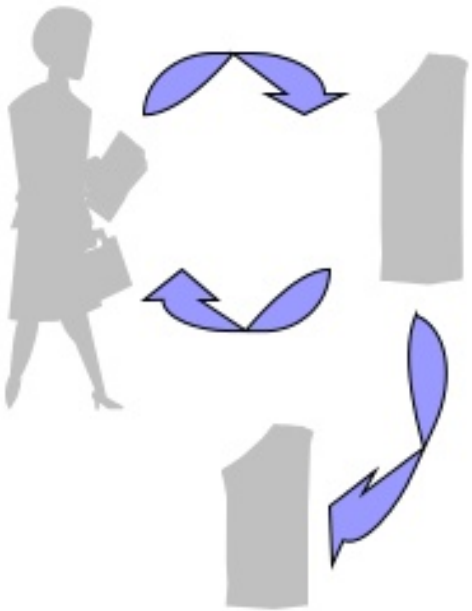

DESIGN PATTERNS
OOP
PROCEDURAL
SEQUENTIAL
PROGRAMMER EVOLUTION

Illustration 1

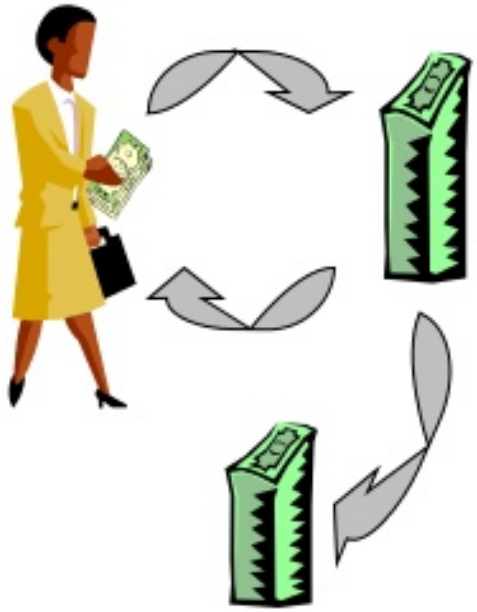
Procedural vs. Object-Oriented

- Procedural



Withdraw, deposit, transfer

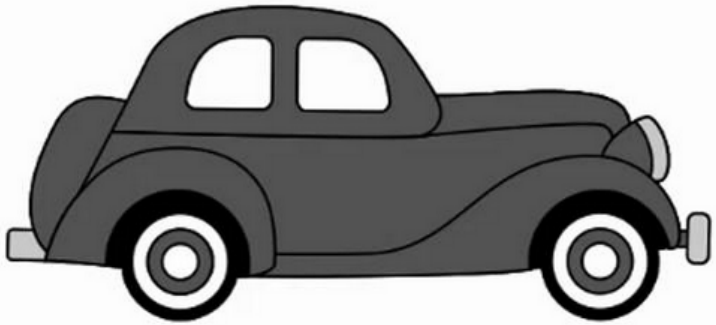
- Object Oriented



Customer, money, account

Illustration 2

Properties
Make
Model
Color
Year
Price



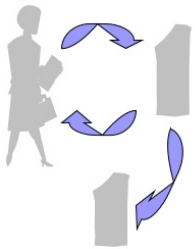
Events
On_Start
On_Parked
On_Brake

Methods
Start
Drive
Park

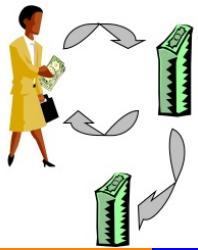
lynda.com

Procedural vs. Object-Oriented

- Procedural



- Object Oriented



Why OOP

- Code reuse !
- Maintainability
- Code reuse !!
- Readability
- Code reuse !!!
- Modularity
- Code reuse !!!!
- Scalability
- Code reuse !!!!!