Module 1 – Fundamentals of Software Design

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Overview and Objectives

## OVERVIEW

As we begin, we will focus on two fundamental concepts in software design: abstraction and exceptions. Abstraction creates a boundary between the application and the client program while exceptions are utilized to handle unplanned events. In this module, you will have an opportunity to recognize these concepts and demonstrate your knowledge to show how they are foundational to software development.

## COURSE LEVEL OBJECTIVES (CLO)

**Upon completion of this course, you should be able to:**

1. Construct modern high quality software systems.
2. Properly define software specifications.
3. Leverage immutability to properly construct concurrent programs.
4. Explain object-oriented concepts such as information hiding, encapsulation, data abstraction, and polymorphism.
5. Properly use exception handling.

## MODULE LEVEL OBJECTIVES (MLO)

**Upon completion of this module’s activities, you should be able to:**

1. Recognize terminology related to specifications, design patterns, and abstraction techniques, including typing, access control, inheritance, and polymorphism. (CLO 2, 4)
2. Illustrate how procedural abstraction can be applied to a real-life situation. (CLO 2, 4)
3. Discuss the advantages of procedural abstraction. (CLO 2, 4)
4. Identify and describe the purpose of at least three types of exceptions. (CLO 5)
5. Write an exception handling example. (CLO 5)

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Module Video (Wiley-Produced w/Dan Ramos) [3-5 minutes]

* Script

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Learning Materials [~150 pages, ~4.5 hours]

## TEXTBOOK READINGS

* Joshua Bloch. Effective Java. Third Edition. Addison-Wesley Professional, 2017, ISBN 978-0-13-468599-1.
  + Chapter 1 and 2
* Barbara Liskov with John Guttag. Program Development in Java. Addison Wesley, 2001, ISBN 0-201-65768-6.
  + Chapter 1 and 2

## ADDITIONAL RESOURCES

* Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... & Werner, L. (2011). [Computational thinking for youth in practice](https://d1wqtxts1xzle7.cloudfront.net/49226520/Computational_thinking_for_youth_in_prac20160929-9740-1f0nmqf-with-cover-page-v2.pdf?Expires=1662580705&Signature=Ipqy94QhiiPsXL8yCyWulFbzXW-R5bSb9MDAOcp6nFoXzbwJlMxQJMwu6vN1fQafGGpyRqAh3mVCFULkAfC88ptrWCCb7Xc7X4CZO2i0uAXnG7Y0NDNcdyS-UoDW6eSMB1O5cYpulnZFzEIaSSg4bEsRMmdEBfxvaAz77o14GZWNuvqfdAYRrBJ8XOhHGOnTNUX~7avTW5dher6tF~qwAcqoxGWNCSlyBwupiz5FuB4pNdj9Eukcb6wpvkQvJ~HqR4aucavThFzx~xhEoKd1Skq7jeRX6iTdKaWYBPbL-JM2PX0qksggINvCiWt8Opkw1lUvI-kwMr7FW4N-6FU8Zw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA). Acm Inroads, 2(1), 32-37.
* Mirolo, C., Izu, C., Lonati, V., & Scapin, E. (2022). [Abstraction in Computer Science Education: An Overview](https://www.infedu.vu.lt/journal/INFEDU/article/720). Informatics in Education, 20(4), 615-639.

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Module 1 Learning Unit 1 – Procedural Abstraction (MLO 1, 2, 3) [~3 hours]

* **Terminology**
  + Dialog Cards:
    - Specifications - <https://csrc.nist.gov/glossary/term/specification>
    - Design patterns – add definition here, cite source
    - Abstraction techniques – add definition here, cite source
    - Typing – add definition here, cite source
    - Access control – add definition here, cite source
    - Inheritance – add definition here, cite source
    - Polymorphism – add definition here, cite source
* **Instructor Screencast**: Introduction to Procedural Abstraction [5 mins, voiceover PPT]
  + Add link to MP4 file or see Module Folders
  + Explain the basic concepts of Procedural Abstraction: history, what it is, what it is not, best practices
* **Interactive Element**: Examples of Procedural Abstraction
  + Real life examples of procedural abstraction
    - In daily life…add content here
    - In organizations and business…add content here
    - Case study example…add content here
* **Interactive Learning**: Understanding the Benefits of Abstraction
  + Use the benefits below to create a Drag the Word activity <https://h5p.org/drag-the-words>
    - It reduces the complexity of viewing things.
    - Avoids code duplication and increases reusability.
    - Helps to increase the security of an application or program as only essential details are provided to the user.
    - It improves the maintainability of the application.
    - It improves the modularity of the application.
    - The enhancement will become very easy because without affecting end-users we can able to perform any type of changes in our internal system.
    - Citation: <https://www.geeksforgeeks.org/abstraction-in-java-2/#:~:text=Advantages%20of%20Abstraction&text=Avoids%20code%20duplication%20and%20increases,the%20modularity%20of%20the%20application>
* **Instructor Screencast**: An Illustrated Example of Procedural Abstraction [5 mins, voiceover PPT]
  + Add link to MP4 file or see Module Folders
  + Show an example of Procedural Abstraction and explain the setup along with the benefits

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Module 1 Learning Unit 2 – Exceptions (MLO 1, 2, 3) [~3 hours]

* **Terminology**
  + Dialog Cards:
    - Runtime errors
    - Try
    - Throw
    - Catch
    - Except
    - Finally
    - Exception Handling
    - Method Overriding
* **Instructor Screencast**: Introduction to Exceptions [5 mins, voiceover PPT]
  + Add link to MP4 file or see Module Folders
  + Explain the basic concepts of Exceptions: history, what they are, what they are not, best practices
* **Interactive Element**: Examples of Exceptions
  + Conceptual knowledge versus implementation of exceptions in real life organizations
    - Articles to read
    - Case studies
    - Examples from industry experience
    - How exceptions vary based on programming language
* **Interactive Learning**: The Basics of Writing Exception Handling
  + Practice Lab – create a practice non-graded tutorial learners can go through to write and fix their own exception handling code (or may there be a practice lab available online?)
* **Instructor Screencast**: Exception Handling Walkthrough [5 mins, voiceover PPT]
  + Add link to MP4 file or see Module Folders
  + Explain a basic exception handling event for learners step by step (give learners enough information and practice in this exercise so they can achieve the module objective to create their own exception handling example for assessment).

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Module 1 Discussion – The Pros and Cons of Procedural Abstraction (MLO 1, 3) [~2.5 hours]

**Purpose**

The purpose of this discussion is to explore the advantages of procedural abstraction.

**Instructions**

Initial post prompt

* List one advantage of using procedural abstraction and explain how the advantage that you highlighted has personally helped in your professional life (if you have not already experienced the benefit personally, consider it might help you in the future).

Reply post prompt

* Provide additional perspective to your peers based on their post. Did they capture the full benefit of the advantage they suggested? How else might it be beneficial to utilize the advantage of procedural abstraction your colleague highlighted? Might there be a con about that aspect of procedural abstraction to take into consideration?

**Deliverable**

Make your initial post (250 words maximum) and respond to at least one peer post (minimum 100 words).

**Due Date**

Your initial post is due by Thursday 11:59pm ET and your reply post(s) by Sunday 11:59pm ET.

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Module 1 Assignment 1 – Procedural Abstraction in Action (MLO 2) [~2.5 hours]

**Purpose**

The purpose of this assignment is to illustrate how procedural abstraction can be applied to a real-life situation.

**Instructions**

Procedural abstraction is a concept that can be applied to many real-life situations. Pick a day to day situation and use it to explain the application of concepts of procedural abstraction (don’t use one that can be found on the internet).

**Deliverable**

Submit a short essay (Word or PDF format, 500 word maximum) with an illustration.

**Due Date**

Your assignment is due by **Sunday 11:59 PM, ET.**

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Module 1 Assignment 2 – Exception Handling Example (MLO 5) [~2.5 hours]

**Purpose**

The purpose of this assignment write an exception handling example.

**Instructions**

Select a coding language of your choice (C++, Java, Python) and write an exception handling example for a given event. You can choose an exception handling approach of your choice (e.g., try…catch block, finally block, etc). You are welcome to describe the setup and event for which you are creating your exception handling as you see fit but keep your scenario within the confines of task related for a business. Along with your example, provide a short tutorial regarding the code you wrote as if you were explaining the concept of exception handling to a novice.

**Deliverable**

Submit the description of the setup/event, code for exception handling and tutorial (Word or PDF format, 500 word maximum).

**Due Date**

Your assignment is due by **Sunday 11:59 PM, ET.**

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Module 1 Quiz (MLO 1) [~.5 hour]

**Purpose**

Quizzes in this course give you an opportunity to demonstrate your knowledge of the subject material.

**Instructions**

Note the following instructions for your quiz:

* The quiz is 20 minutes in length.
* The quiz is closed-book.

**Deliverable**

Use the link above to take the quiz.

**Due Date**

Your quiz submission is due by Sunday 11:59 PM, ET.

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_