



## ĐỀ CƯƠNG MODULE ADVANCED PROGRAMMING WITH JAVA

**Mã số:** BC-JAVA-APJ

**Version:** 2.0

**Ngày cập nhật:** 20/12/2019

### 1. Giới thiệu tổng quan

Module Advanced Programming with Java giúp học viên nắm vững các khái niệm và kỹ thuật cốt lõi trong lập trình, nâng cao tư duy và kỹ năng lập trình. Kết thúc module này, học viên thành thạo việc phát triển các ứng dụng dựa trên ngôn ngữ Java, mô hình lập trình Hướng đối tượng và Cấu trúc dữ liệu và giải thuật.

### 2. Mục tiêu

Kết thúc module học viên thành thạo được các kỹ thuật lập trình với ngôn ngữ Java và xây dựng được các ứng dụng theo mô hình Lập trình Hướng đối tượng, xây dựng được các ứng dụng sử dụng ngôn ngữ lập trình Java. Kết thúc module, học viên đủ năng lực để học lập trình web với ngôn ngữ Java.

#	Mục tiêu
1.1.9	Thành thạo cú pháp của ngôn ngữ Java
1.3.1	Trình bày được mô hình Lập trình Hướng đối tượng
1.3.4	Sử dụng được UML
1.6.1	Tuân thủ các nguyên lý căn bản trong thiết kế
1.4.1	Sử dụng được các cấu trúc dữ liệu danh sách
1.4.2	Sử dụng được các cấu trúc dữ liệu Stack, Queue, Map và Tree
1.4.4	Triển khai được các giải thuật cơ bản
1.5.1	Tuân thủ Clean Code
1.5.2	Sử dụng được các kỹ thuật refactoring
1.9.5	Triển khai được kiểm thử tự động
1.2.1	Sử dụng được try-catch, xử lý được ngoại lệ
1.2.5	Thao tác được với file
1.2.6	Triển khai được Threading

1.2.2	Debug được ứng dụng
1.6.2	Trình bày được ý nghĩa của Design Pattern

### 3. Chuẩn đầu ra

Mã số	Chuẩn đầu ra
<b>1.1.9</b>	<b>Thành thạo cú pháp ngôn ngữ Java</b>
1.1.9.1	Sử dụng được biến trong Java
1.1.9.2	Sử dụng được kiểu dữ liệu trong Java
1.1.9.3	Sử dụng được toán tử trong Java
1.1.9.4	Sử dụng được câu lệnh if-else trong Java
1.1.9.5	Sử dụng được câu lệnh switch-case trong Java
1.1.9.6	Sử dụng được câu lệnh for trong Java
1.1.9.7	Sử dụng được câu lệnh while trong Java
1.1.9.8	Sử dụng được câu lệnh do-while trong Java
1.1.9.9	Sử dụng được câu lệnh break trong Java
1.1.9.10	Sử dụng được câu lệnh continue trong Java
1.1.9.11	Khai báo được mảng trong Java
1.1.9.12	Duyệt được mảng trong Java
1.1.9.13	Trình bày được nền tảng Java
1.1.9.14	Tạo và thực thi được ứng dụng Java
1.1.9.15	Thực hiện được các thao tác với mảng trong Java
1.1.9.16	Sử dụng được phương thức trong Java
<b>1.3.1</b>	<b>Trình bày được mô hình Lập trình Hướng đối tượng</b>
1.3.1.1	Trình bày được khái niệm mô hình hoá trong OOP
1.3.1.2	Trình bày được 4 đặc trưng của OOP
1.3.1.3	Phân biệt được lớp và đối tượng
1.3.1.4	Thiết kế được các giải pháp cơ bản sử dụng theo mô hình Hướng Đối tượng (*)
<b>1.3.4</b>	<b>Sử dụng được UML</b>
1.3.4.1	Sử dụng được các ký hiệu UML để mô tả lớp, interface và các mối quan hệ
1.3.4.2	Sử dụng được các ký hiệu UML để mô tả biểu đồ activity (*)
<b>1.6.1</b>	<b>Tuân thủ các nguyên lý căn bản trong thiết kế</b>
1.6.1.1	Hiểu được ý nghĩa của SOLID trong thiết kế (*)
1.6.1.2	Tuân thủ nguyên lý Single Responsibility trong thiết kế (*)
1.6.1.3	Áp dụng tốt tư duy Simple Design (**)
1.6.1.4	Sử dụng được các kỹ thuật refactor để có thiết kế tốt hơn (**)
<b>1.4.1</b>	<b>Sử dụng được các cấu trúc dữ liệu danh sách</b>
1.4.1.1	Sử dụng được ArrayList
1.4.1.2	Sử dụng được LinkedList
1.4.1.3	Sử dụng được Set
1.4.1.4	Phân biệt được các trường hợp sử dụng List, LinkedList và Set (*)
1.4.1.5	Triển khai được ArrayList (*)

1.4.1.6	Triển khai được LinkedList (*)
1.4.1.7	Triển khai được Set (**)
<b>1.4.2</b>	<b>Sử dụng được các cấu trúc dữ liệu Stack, Queue, Map và Tree</b>
1.4.2.1	Phân biệt được các trường hợp sử dụng Stack, Queue, Map và Tree (**)
1.4.2.2	Sử dụng được cấu trúc dữ liệu Stack
1.4.2.3	Triển khai được Stack (**)
1.4.2.4	Sử dụng được cấu trúc dữ liệu Queue
1.4.2.5	Triển khai được Queue (**)
1.4.2.6	Sử dụng được cấu trúc Map
1.4.2.7	Sử dụng được cấu trúc Tree (*)
<b>1.4.4</b>	<b>Triển khai được các giải thuật cơ bản</b>
1.4.4.1	Triển khai được thuật toán sắp xếp nổi bọt
1.4.4.2	Triển khai được thuật toán sắp xếp chèn
1.4.4.3	Triển khai được thuật toán sắp xếp chọn
1.4.4.4	Triển khai được Thuật toán tìm kiếm tuyến tính
1.4.4.5	Triển khai được Thuật toán tìm kiếm nhị phân (*)
1.4.4.6	Thao tác được với chuỗi
1.4.4.7	Sử dụng được Regular Expression
1.4.4.8	Thành thạo Regular Expression (*)
<b>1.5.1</b>	<b>Tuân thủ Clean Code</b>
1.5.1.1	Trình bày được tầm quan trọng của Clean Code
1.5.1.2	Tuân thủ Baby steps
1.5.1.3	Tuân thủ Coding Convention
<b>1.5.2</b>	<b>Sử dụng được các kỹ thuật refactoring</b>
1.5.2.1	Sử dụng được các kỹ thuật đổi tên
1.5.2.2	Sử dụng được kỹ thuật tách biến
1.5.2.3	Sử dụng được kỹ thuật tách phương thức
1.5.2.4	Sử dụng được kỹ thuật tách hằng
1.5.2.5	Sử dụng được kỹ thuật tách lớp (*)
<b>1.9.5</b>	<b>Triển khai được kiểm thử tự động</b>
1.9.5.1	Triển khai được kiểm thử đơn vị (*)
1.9.5.2	Triển khai được kiểm thử tích hợp (**)
1.9.5.3	Triển khai được cơ chế mocking (**)
<b>1.2.1</b>	<b>Sử dụng được try-catch, xử lý được ngoại lệ</b>
1.2.1.1	Sử dụng được khối lệnh try-catch
1.2.1.2	Sử dụng được khối lệnh finally
1.2.1.3	Sử dụng đúng loại Exception
1.2.1.4	Khai báo được lớp Exception tùy biến (*)
<b>1.2.5</b>	<b>Thao tác được với file</b>
1.2.5.1	Thực hiện được thao tác cơ bản với file và thư mục
1.2.5.2	Thực hiện được các thao tác đọc và ghi file text
1.2.5.3	Thực hiện được các thao tác đọc và ghi file nhị phân

1.2.5.4	Triển khai được cơ chế Serialization trong java
<b>1.2.6</b>	<b>Triển khai được Threading</b>
1.2.6.1	Triển khai được cơ chế multi-threading
1.2.6.2	Triển khai được cơ chế đồng bộ trong các ứng dụng multi-threading
<b>1.2.2</b>	<b>Debug được ứng dụng</b>
1.2.2.1	Sử dụng được trình debugger của IDE
1.2.2.2	Debug được theo phương pháp thử công
<b>1.6.2</b>	<b>Sử dụng được Design Pattern cơ bản</b>
1.6.2.1	Trình bày được lợi ích của Design Pattern
1.6.2.2	Liệt kê được 5 loại Design Pattern thông dụng (*)
1.6.2.3	Triển khai được Singleton Pattern (*)
1.6.2.4	Triển khai được Factory Method Pattern (**)
1.6.2.5	Triển khai được Object Pool Pattern (**)
1.6.2.6	Triển khai được Adapter Pattern (**)
1.6.2.7	Triển khai được Facade Pattern (**)
1.6.2.8	Triển khai được Proxy Pattern (**)
1.6.2.9	Triển khai được Command Pattern (**)
1.6.2.10	Triển khai được Observer Pattern (**)
1.6.2.11	Triển khai được Template Method Pattern (**)
1.6.2.12	Triển khai được Strategy Pattern (**)

## 4. Tài liệu học tập

### 4.1. Tài liệu cho học viên

- CodeGymX: [\[BC-JAVA-APJ\] Advanced Programming with Java 2.0](#)
- CodeGymX: [Khóa học CSCD 2](#)
- CodeGymP: Hệ thống quản lý dự án
  - [Backlog 8 x 5](#)
  - [Backlog 4 X5](#)
- [Hệ thống mã nguồn có sẵn](#)
- Sách “Introduction to Java\_Programming\_Liang”

### 4.2. Tài liệu cho Trainer

- Slide bài giảng
- Teacher Guide
- CodeGymX: [\[BC-JAVA-APJ\] Advanced Programming with Java 2.0](#)
- CodeGymX: [Khóa học CSCD 2](#)
- CodeGymP: Hệ thống quản lý dự án
  - [Backlog 8 x 5](#)
  - [Backlog 4 X5](#)
- [Hệ thống mã nguồn có sẵn](#)
- CodeGymS: [Hệ thống quản lý học viên](#)

### 4.3. Tài liệu tham khảo

- Introduction to Java\_Programming\_Liang
- Design Pattern - Source Making

## 5. Đánh giá

### 5.1 Theo dõi và Đánh giá trong quá trình học

Các cơ chế theo dõi và đánh giá trong quá trình học bao gồm:

- Phản hồi nhanh hằng ngày:
  - Tutor chịu trách nhiệm hỗ trợ, giải đáp các vướng mắc của học viên trong giờ tự học để giúp học viên hoàn thành các nhiệm vụ học tập theo kế hoạch
  - Tutor và Coach cập nhật thông tin về các giờ dạy hàng ngày bằng cơ chế ghi log trên CodeGymS hoặc daily meeting (5 - 15'/ngày)
  - Tutor và Huấn luyện viên phản hồi với từng học viên khi phát hiện có vấn đề cần can thiệp, điều chỉnh
- Học viên làm Quiz từng bài học
- Kết thúc tuần, học viên gửi Báo cáo công việc tuần và Lập kế hoạch công việc cho tuần tiếp theo theo mẫu và gửi đến Huấn luyện viên, Đại diện đơn vị sử dụng lao động (nếu là học viên do doanh nghiệp cử đi), cc cho Tutor và giáo vụ.

### 5.2. Cấu trúc đánh giá

#	Hạng mục đánh giá	Thời lượng	Thang điểm	Điểm đạt	Hình thức kiểm tra	Cách thức đánh giá
1	Thi lý thuyết	40 phút	100	$\geq 75$	Trắc nghiệm	Tổ chức thi trắc nghiệm sử dụng đề thi được cung cấp.
2	Thi thực hành	120 phút	100	$\geq 75$	Thực hành	Tổ chức thi thực hành sử dụng đề thi được cung cấp.
3	Đánh giá năng lực	-	-	100% các năng lực bắt buộc	-	- Học viên tự đánh giá năng lực bản thân sử dụng bản Tự đánh giá năng lực. - Huấn luyện viên đánh giá năng lực học viên dựa trên rubric.

### Lưu ý:

Các mức năng lực bao gồm:

- **Không đạt:** Học viên chưa sẵn sàng để đảm nhận các công việc cần đến năng lực đang được đánh giá
- **Đạt:** Học viên có thể tham gia các công việc cần đến năng lực đang được đánh giá, với tốc độ chậm, hiệu quả chưa cao
- **Tốt:** Học viên có thể tham gia các công việc cần đến năng lực đang được đánh giá, với tốc độ và hiệu quả trung bình, có khả năng tiến bộ nhanh
- **Xuất sắc:** Học viên có thể tham gia các công việc cần đến năng lực đang được đánh giá với tốc độ và hiệu quả cao. Năng lực này là thế mạnh của học viên

## 6. Yêu cầu

### 6.1. Yêu cầu cơ sở vật chất

- Máy chiếu/tivi
- Bảng và bút viết

### 6.2. Yêu cầu cấu hình máy tính

- Hệ điều hành: 64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+ | OS X 10.9 (Marveric)+
- Trình duyệt: IE11/Edge | Chrome 60+ | Firefox 55+
- IDE: IntelliJ IDEA
- Git 2.14+

### 6.3. Yêu cầu chương trình học

- Đã hoàn thành module BootCamp Preparation hoặc đạt trình độ tương đương

## 7. Thời gian

	Full-time	Part-time
Lý thuyết	34.5	34.5
Thực hành	149.5	57.5
Tự học	69	138
Đánh giá	3	3
Tổng thời lượng	256	233

## 8. Cách thức học module

Học viên cần sắp xếp kế hoạch ngày để thực hiện các task màu vàng của backlog module trên CodeGymP trước khi vào giờ học lý thuyết. Sau giờ lý thuyết, hoàn thành các task màu xanh (task bắt buộc) và màu cam (task khuyến khích làm để thuần thực kỹ năng).

## 9. Phân bố chương trình

Session <i>Bài học</i>	Topic / Learning Outcome <i>Chủ đề bài học/Chuẩn đầu ra bài học</i>	Objective <i>Mục tiêu</i>
1 APJ.T1 APJ.L1	<b>Introduction to Java</b> <ul style="list-style-type: none"> <li>• Trình bày được nội dung, yêu cầu, lịch trình và kết quả của module Advanced Programming with Java</li> <li>• Sử dụng được cú pháp Java để thao tác với biến</li> <li>• Sử dụng được cú pháp Java để thao tác với cấu trúc điều kiện</li> </ul>	1.1.9.1 1.1.9.2 1.1.9.3 1.1.9.4 1.1.9.5 1.1.9.9 1.1.9.10 1.1.9.13 1.1.9.14
2 APJ.T2 APJ.L2	<b>Vòng lặp và mảng trong Java</b> <ul style="list-style-type: none"> <li>• Sử dụng được cú pháp Java để thao tác với cấu trúc lặp</li> <li>• Sử dụng được cú pháp Java để thao tác với mảng</li> </ul>	1.1.9.6 1.1.9.7 1.1.9.8 1.1.9.9 1.1.9.10 1.1.9.11 1.1.9.12 1.1.9.13 1.1.9.14 1.1.9.15
3 APJ.T3 APJ.L3	<b>Phương thức trong Java</b> <ul style="list-style-type: none"> <li>• Sử dụng được phương thức trong Java</li> </ul>	1.1.9.16
4 APJ.T4 APJ.L4	<b>Lớp và đối tượng trong Java</b> <ul style="list-style-type: none"> <li>• Trình bày được mô hình lập trình hướng đối tượng</li> <li>• Trình bày được các khái niệm lớp, đối tượng, phương thức, thuộc tính, hàm tạo</li> <li>• Trình bày được cú pháp khai báo lớp</li> <li>• Trình bày được cú pháp khởi tạo đối tượng</li> <li>• Trình bày được cách truy xuất thuộc tính, phương thức của lớp</li> <li>• Tạo và sử dụng được các đối tượng đơn giản</li> <li>• Mô tả được lớp bằng biểu đồ</li> </ul>	1.3.1.1 1.3.1.2 1.3.1.3 1.3.1.4 1.3.2.1 1.3.2.2 1.3.2.3 1.3.2.4 1.3.2.5 1.3.2.9 1.3.4.1 1.3.4.2

5 APJ.T5 APJ.L5	<b>Access modifier, static method, static property</b> <ul style="list-style-type: none"> <li>• Phân biệt được biến kiểu dữ liệu nguyên thuỷ và biến tham chiếu</li> <li>• Phân biệt được biến của lớp và biến của đối tượng</li> <li>• Phân biệt được phương thức của lớp và phương thức của đối tượng</li> <li>• Khai báo và sử dụng được các biến Static</li> <li>• Khai báo và sử dụng được các phương thức Static</li> <li>• Trình bày được cú pháp khai báo Package</li> <li>• Trình bày được các Access Modifier</li> </ul>	1.3.2.6 1.3.2.7
6 APJ.T6 APJ.L6 CSCD2.L1	<b>Kế thừa</b> <ul style="list-style-type: none"> <li>• Trình bày được cơ chế kế thừa</li> <li>• Triển khai được cơ chế kế thừa giữa các lớp</li> <li>• Trình bày được cơ chế ghi đè phương thức (Method Overriding)</li> <li>• Biểu diễn được mối quan hệ kế thừa bằng các kí hiệu</li> <li>• Trình bày được ý nghĩa của từ khoá final</li> <li>• Trình bày được khái niệm Polymorphism</li> <li>• Trình bày được phương thức toString() của lớp Object</li> <li>• Trình bày được cơ chế ép kiểu (Casting)</li> </ul>	1.3.3.1 1.3.3.2 1.3.3.3 1.3.3.4
	<b>CSCD: Đọc sách</b>	3.1.2.1 3.1.2.2 3.1.2.4
7 APJ.T7 APJ.L7	<b>Abstract Class &amp; Interface</b> <ul style="list-style-type: none"> <li>• Trình bày được Interface</li> <li>• Trình bày được Abstract Class</li> <li>• Trình bày được Abstract Method</li> <li>• Khai báo được Interface</li> <li>• Khai báo được Abstract Class</li> <li>• Khai báo được lớp triển khai từ Interface</li> <li>• Khai báo được lớp kế thừa từ Abstract Class</li> <li>• Thiết kế được các giải pháp có sử dụng Interface và Abstract Class</li> <li>• Trình bày và sử dụng được Anonymous Class</li> </ul>	1.3.3.3 1.3.3.4



8 APJ.T8 APJ.L8	<b>Clean Code &amp; Refactoring</b> <ul style="list-style-type: none"> <li>• Trình bày được Clean Code</li> <li>• Trình bày được các tiêu chí cốt lõi của Clean Code</li> <li>• Nhận diện được các mã nguồn bẩn cơ bản</li> <li>• Thực hiện được kĩ thuật đổi tên biến</li> <li>• Thực hiện được kĩ thuật đổi tên phương thức</li> <li>• Thực hiện được kĩ thuật tách biến</li> <li>• Thực hiện được kĩ thuật tách hằng</li> <li>• Thực hiện được kĩ thuật tách phương thức</li> </ul>	1.5.1.1 1.5.1.2 1.5.1.3 1.5.2.1 1.5.2.2 1.5.2.3 1.5.2.4 1.5.2.5
9 APJ.T9 APJ.L9	<b>Automation test &amp; TDD</b> <ul style="list-style-type: none"> <li>• Trình bày được tư duy Test-First</li> <li>• Trình bày được mô hình TDD</li> <li>• Mô tả được vòng vận hành của TDD</li> <li>• Trình bày được hoạt động của Refactoring và lợi ích của Refactoring</li> <li>• Trình bày được khái niệm UnitTest</li> <li>• Giải thích cách viết TestCase cơ bản</li> <li>• Triển khai được UnitTest với Junit</li> <li>• Áp dụng tư duy Test-First vào code</li> <li>• Viết được UnitTest</li> <li>• Chạy được UnitTest</li> </ul>	2.2.1.1 2.2.1.2 2.2.1.3 2.2.2.1 2.2.2.2 2.2.2.3 2.2.2.4 2.2.2.5 2.2.2.6
10 APJ.T10 APJ.L10	<b>DSA: Danh sách, Generic</b> <ul style="list-style-type: none"> <li>• Trình bày được khái niệm danh sách</li> <li>• Trình bày được khái niệm danh sách mảng</li> <li>• Cài đặt được danh sách mảng</li> <li>• Trình bày được khái niệm danh sách liên kết</li> <li>• Cài đặt được danh sách liên kết</li> <li>• Trình bày được khái niệm Generics</li> <li>• Triển khai được cơ chế Generics</li> </ul>	1.4.1.1 1.4.1.2 1.4.1.3 1.4.1.4 1.4.1.5 1.4.1.6 1.4.1.7 1.3.5.1 1.3.5.2

11 APJ.T11 APJ.L11	<b>DSA: Stack, Queue, Map, Tree</b> <ul style="list-style-type: none"> <li>• Trình bày được cấu trúc dữ liệu Stack</li> <li>• Cài đặt được cấu trúc dữ liệu Stack</li> <li>• Trình bày được cấu trúc dữ liệu Queue</li> <li>• Cài đặt được cấu trúc dữ liệu Queue</li> <li>• Trình bày được cấu trúc dữ liệu Map</li> <li>• Sử dụng được Map trong Java Collection Framework</li> <li>• Trình bày được cấu trúc dữ liệu Tree</li> <li>• Triển khai được cấu trúc dữ liệu Tree</li> </ul>	1.4.2.1 1.4.2.2 1.4.2.3 1.4.2.4 1.4.2.5 1.4.2.6 1.4.2.7
12 APJ.T12 APJ.L12	<b>Java Collection Framework</b> <ul style="list-style-type: none"> <li>• Trình bày được Java Collection Framework</li> <li>• Phân biệt được các trường hợp sử dụng của ArrayList, LinkedList và Set</li> <li>• Sử dụng được cấu trúc dữ liệu ArrayList</li> <li>• Sử dụng được cấu trúc dữ liệu LinkedList</li> <li>• Sử dụng được cấu trúc dữ liệu Set</li> </ul>	1.4.1.1 1.4.1.2 1.4.1.3 1.4.1.4 1.4.1.5 1.4.1.6 1.4.1.7
13 APJ.T13 APJ.L13	<b>Thuật toán tìm kiếm</b> <ul style="list-style-type: none"> <li>• Trình bày được thuật toán tìm kiếm tuyến tính</li> <li>• Cài đặt được thuật toán tìm kiếm tuyến tính</li> <li>• Trình bày được thuật toán tìm kiếm nhị phân</li> <li>• Cài đặt được thuật toán tìm kiếm nhị phân</li> <li>• Trình bày được độ phức tạp của thuật toán</li> <li>• Tính được độ phức tạp của thuật toán cho những trường hợp thông dụng</li> </ul>	1.4.4.4 1.4.4.5 1.4.3.1 1.4.3.2
14 APJ.T14 APJ.L14	<b>Thuật toán sắp xếp</b> <ul style="list-style-type: none"> <li>• Trình bày được thuật toán sắp xếp nổi bọt</li> <li>• Cài đặt được thuật toán sắp xếp nổi bọt</li> <li>• Trình bày được thuật toán sắp xếp chọn</li> <li>• Cài đặt được thuật toán sắp xếp chọn</li> <li>• Trình bày được thuật toán sắp xếp chèn</li> <li>• Cài đặt được thuật toán sắp xếp chèn</li> </ul>	1.4.4.1 1.4.4.2 1.4.4.3
15 APJ.T15 APJ.L15	<b>Xử lý ngoại lệ và Debug</b> <ul style="list-style-type: none"> <li>• Trình bày được cơ chế của ngoại lệ</li> <li>• Sử dụng được cấu trúc try..catch..finally</li> <li>• Sử dụng được trình Debugger của IDE</li> </ul>	1.2.1.1 1.2.1.2 1.2.1.3 1.2.1.4

	<ul style="list-style-type: none"> <li>• Debug được theo phương pháp thủ công</li> </ul>	1.2.2.1 1.2.2.2
16 APJ.T16 APJ.L16	<b>IO: Text File</b> <ul style="list-style-type: none"> <li>• Trình bày được gói java.io</li> <li>• Trình bày được Stream</li> <li>• Thực hiện được thao tác cơ bản với File và thư mục</li> <li>• Thực hiện được các thao tác đọc và ghi với file text</li> </ul>	1.2.5.1 1.2.5.2
17 APJ.T17 APJ.L17 CSCD2.L2	<b>IO: Binary File &amp; Serialization</b> <ul style="list-style-type: none"> <li>• Trình bày được các lựa chọn khác nhau để thao tác với file nhị phân</li> <li>• Thực hiện được các thao tác đọc và ghi file nhị phân</li> <li>• Trình bày được khái niệm Serialization</li> </ul>	1.2.5.3 1.2.5.4
	<b>CSCD: Đọc sách</b> <b>CSCD: Learning How to learn</b>	3.1.2.1 3.1.2.2 3.1.2.4
18 APJ.T18 APJ.L18	<b>Threading</b> <ul style="list-style-type: none"> <li>• Trình bày được cơ chế hoạt động của Thread</li> <li>• Trình bày được cơ chế Multi-Threading</li> <li>• Triển khai được cơ chế Multi-Threading</li> <li>• Triển khai được cơ chế đồng bộ trong các ứng dụng Multi-Threading</li> </ul>	1.2.6.1 1.2.6.2
19 APJ.T19 APJ.L19	<b>String &amp; Regex</b> <ul style="list-style-type: none"> <li>• Trình bày được chuỗi trong java</li> <li>• Sử dụng được các phương thức xử lý chuỗi có sẵn trong java</li> <li>• Sử dụng được lớp StringBuilder</li> <li>• Trình bày được Regular Expression và các ứng dụng của Regular Expression</li> <li>• Trình bày được các Character Class trong java</li> <li>• Trình bày được các Quantifier trong java</li> <li>• Sử dụng Regular Expression để validate chuỗi, tìm kiếm trong chuỗi, crawl dữ liệu text</li> </ul>	1.4.4.6 1.4.4.7 1.4.4.8
20 APJ.T20 APJ.L20	<b>OOAD &amp; SOLID</b> <ul style="list-style-type: none"> <li>• Hiểu được ý nghĩa của SOLID trong thiết kế</li> <li>• Tuân thủ nguyên lý Single Responsibility trong thiết kế</li> </ul>	1.6.1.1 1.6.1.2 1.6.1.3

	<ul style="list-style-type: none"> <li>• Áp dụng tốt tư duy Simple Design</li> <li>• Sử dụng được các kĩ thuật refactor để có thiết kế tốt hơn</li> </ul>	1.6.1.4
21 APJ.T21 APJ.L21	<b>Design Patterns: Design Patterns</b> <ul style="list-style-type: none"> <li>• Trình bày được ý nghĩa của Design Pattern</li> <li>• Trình bày được ý nghĩa của các Creational Design Pattern</li> <li>• Triển khai được Singleton Pattern</li> <li>• Triển khai được Factory Method Pattern</li> <li>• Triển khai được Object Pool Pattern</li> </ul>	1.6.2.1 1.6.2.2 1.6.2.3 1.6.2.4 1.6.2.5
22 APJ.T22 APJ.L22	<b>Design Patterns: Structural Design Pattern</b> <ul style="list-style-type: none"> <li>• Trình bày được ý nghĩa của các Structural Design Pattern</li> <li>• Triển khai được Adapter Design Pattern</li> <li>• Triển khai được Facade Design Pattern</li> <li>• Triển khai được Proxy Design Pattern</li> </ul>	1.6.2.6 1.6.2.7 1.6.2.8
23 APJ.T24 APJ.L24 CSCD2.T3 CSCD2.L3	<b>Design Patterns: Behavioral Design Pattern</b> <ul style="list-style-type: none"> <li>• Trình bày được ý nghĩa của các Behavioral Design Pattern</li> <li>• Triển khai được Template Method Design Pattern</li> <li>• Triển khai được Command Design Pattern</li> <li>• Triển khai được Observer Design Pattern</li> <li>• Triển khai được Strategy Design Pattern</li> </ul>	1.6.2.9 1.6.2.10 1.6.2.11 1.6.2.12
	<b>CSCD: Đọc sách</b> <b>CSCD: Seminar – Cách mạng công nghiệp 4.0</b> <b>CSCD: Viết reflection về cách học lập trình hiệu quả</b>	3.1.2.1 3.1.2.2 3.1.2.4 3.1.3.2
	<b>CSCD: Đọc sách</b> <b>CSCD: Seminar Xu hướng công nghệ của năm</b> <b>CSCD: Retrospective cá nhân về module</b>	3.1.2.1 3.1.2.2 3.1.2.4 3.1.3.1 3.2.1.2 3.3.1.3
24 APJ.Exam	<b>Đánh giá cuối module</b> Thi lý thuyết và thực hành Học viên tự đánh giá năng lực Coach đánh giá năng lực học viên	