Java Programming II

Course Syllabus

📋 Course Information

Course Code: COMP 3192
Course Title: Java Programming II
Duration: 16 Weeks
Format: Online
Schedule: One day per week, 2 -3 hours per session
Start Date: Every Saturday (6:30pm) From 20-12-2025
Prerequisites: Basic programming knowledge (especially Java)
Batch: 25
Semester: 6
ចំនួនក្រេឌីត៖ ៣ = ៤៥ម៉ោង (សិក្សាទ្រឹស្តី ២២.៥ ម៉ោង និងអនុវត្ត ២២.៥ ម៉ោង)

👨‍🏫 Instructor Information

Instructor: SET YUTH
Telegram: https://t.me/setyuth
Email: set.yuth@gmail.com
Response Time: Within 24-48 hours
Communication: Telegram, IIC E-Learning Forum, Google Meet

📖 Course Description

This Advanced Java course is designed for students with varying programming backgrounds, from beginners to those with some Java experience. The course takes a practical, hands-on approach to learning Java, starting with fundamental concepts and progressively building toward enterprise-level web application development.

By the end of this course, students will be proficient in Object-Oriented Programming, database integration, and modern web application development using Spring Boot. The course culminates in a capstone project where students will build a complete web application with database connectivity.

🎯 Course Objectives

Upon successful completion of this course, students will be able to:

1. **Master Java Fundamentals:** Write clean, efficient Java code using proper syntax, control structures, and methods
2. **Apply Object-Oriented Programming:** Design and implement robust applications using OOP principles (encapsulation, inheritance, polymorphism, abstraction)
3. **Work with Collections:** Utilize Java Collections Framework for data management

4. **Handle Exceptions:** Implement proper error handling and file I/O operations
5. **Database Integration:** Design database schemas and integrate PostgreSQL with Java applications using JDBC
6. **Build Web Applications:** Create RESTful web services using Spring Boot framework
7. **Implement Data Persistence:** Use Spring Data JPA for database operations
8. **Develop Complete Applications:** Build, test, and deploy full-stack applications with proper architecture

---

💻 Required Software & Tools

All software is free and open-source:

1. **Java Development Kit (JDK) 17 or later**
    - Download: https://adoptium.net/ or https://www.oracle.com/java/technologies/downloads/
2. **Integrated Development Environment (IDE)**
    - IntelliJ IDEA Community Edition (Recommended)
    - Download: https://www.jetbrains.com/idea/download/
    - Alternative: Eclipse IDE
3. **PostgreSQL Database (Version 14 or later)**
    - Download: https://www.postgresql.org/download/
4. **API Testing Tool**
    - Postman
    - Download: https://www.postman.com/downloads/
5. **Version Control (Optional but Recommended)**
    - Git
    - Download: https://git-scm.com/downloads
6. **Google Meet**
    - For online classes
    - Download: https://meet.google.com/

📚 Course Materials

Required Resources (All Free):

- **Course Notes:** Provided weekly via course platform
- **Official Java Documentation:** https://docs.oracle.com/javase/tutorial/
- **Spring Boot Documentation:** https://spring.io/projects/spring-boot
- **PostgreSQL Documentation:** https://www.postgresql.org/docs/

Recommended (Optional):

- *Effective Java* by Joshua Bloch
- *Head First Java* by Kathy Sierra & Bert Bates
- *Spring Boot in Action* by Craig Walls
- Online platform: CodingBat (https://codingbat.com/java) for practice

---

📅 Weekly Course Schedule

Week 1: Java Fundamentals Review & Setup

Topics:

- Java development environment setup (JDK, IDE, Maven)
- Variables, data types, and operators
- Control structures (if-else, switch, loops)
- Methods and parameter passing
- User input with Scanner class

Hands-on Project: Simple Calculator Application

Deliverables:

- Working calculator with basic operations
- Homework: Enhanced calculator with additional features

Week 2: Object-Oriented Programming Essentials

Topics:

- Classes and Objects
- Constructors (default and parameterized)
- Method overloading
- Encapsulation and access modifiers (public, private, protected)
- Getters and setters with validation
- The `this` keyword

Hands-on Project: Student Management System (Console-based)

Deliverables:

- Functional CRUD application
- Homework: Add Student ID and additional features

Week 3: Inheritance and Polymorphism

Topics:

- Inheritance and the `extends` keyword

- The `super` keyword
- Method overriding vs method overloading
- Runtime polymorphism
- Abstract classes and methods
- The `instanceof` operator and casting

Hands-on Project: University Management System (Students, Teachers, Admins)

Deliverables:

- Multi-user type management system
- Homework: Add Graduate Student class

Week 4: Interfaces and Java Collections Framework

Topics:

- Interfaces and the `implements` keyword
- Abstract classes vs interfaces
- Multiple inheritance through interfaces
- Introduction to Collections Framework
- List interface: ArrayList, LinkedList
- Set interface: HashSet, TreeSet
- Map interface: HashMap, TreeMap
- Iterating through collections

Hands-on Project: Enhanced Student System with Collections

Deliverables:

- Refactored system using appropriate collections
- Homework: Implement sorting and searching

Week 5: Exception Handling and File I/O

Topics:

- Understanding exceptions and errors
- Try-catch-finally blocks
- Throwing and creating custom exceptions
- Exception hierarchy
- Reading and writing text files
- Working with CSV files
- BufferedReader and BufferedWriter
- Try-with-resources statement

Hands-on Project: Data Persistence with File I/O

Deliverables:

- Student system with file-based storage
- Homework: CSV import/export functionality

Week 6: Introduction to Databases and SQL

Topics:

- Database concepts and terminology
- Relational database design
- PostgreSQL installation and setup
- Basic SQL commands (SELECT, INSERT, UPDATE, DELETE)
- WHERE clauses and filtering
- JOIN operations (INNER, LEFT, RIGHT)
- Creating tables and defining relationships
- Primary keys and foreign keys
- Data types in PostgreSQL

Hands-on Project: Database Schema Design

Deliverables:

- Complete database schema for student system
- Homework: Write SQL queries for common operations

Week 7: JDBC - Connecting Java to PostgreSQL

Topics:

- Introduction to JDBC (Java Database Connectivity)
- JDBC architecture and drivers
- Setting up PostgreSQL JDBC driver
- Establishing database connections
- Connection strings and configuration
- Executing SQL statements (Statement vs PreparedStatement)
- Working with ResultSet
- SQL injection prevention
- Transaction management
- Connection pooling basics

Hands-on Project: Student System with Database Backend

Deliverables:

- Console application with full JDBC integration
- Homework: Add transaction support

Week 8: Mid-Course Project Week

Goal: Consolidate all learning from Weeks 1-7

Project: Library Management System

- Complete console-based CRUD application
- Object-oriented design with inheritance
- Database integration with PostgreSQL
- Exception handling and validation
- File I/O for backup/restore

Requirements:

- Books, Members, and Borrowing records
- Search and filter functionality
- Due date tracking
- Report generation

Deliverables:

- Complete working application
- Project documentation
- Source code with comments

Assessment: 20% of final grade

Week 9: Introduction to Web Development Concepts

Topics:

- Client-Server architecture
- HTTP protocol fundamentals (GET, POST, PUT, DELETE)
- REST API principles
- JSON format and data exchange
- Introduction to Spring Framework ecosystem
- What is Spring Boot and its advantages
- Spring Boot project structure
- Dependency injection basics

Hands-on Project: First Spring Boot Application

Deliverables:

- Hello World REST API
- Homework: Add multiple endpoints

Week 10: Spring Boot Basics

Topics:

- Spring Boot project structure explained

- Important annotations (@SpringBootApplication, @RestController, @GetMapping)
- Creating REST endpoints
- Path variables and request parameters
- Request and Response bodies
- HTTP status codes
- Application configuration (application.properties)
- Running and testing Spring Boot applications
- Using Postman for API testing

Hands-on Project: Simple REST API (Book Management)

Deliverables:

- REST API with CRUD operations (in-memory)
- Homework: Add validation

Week 11: Spring Boot with Database Integration

Topics:

- Introduction to Spring Data JPA
- ORM (Object-Relational Mapping) concepts
- Entity classes and annotations (@Entity, @Id, @GeneratedValue)
- Repository pattern and JpaRepository
- Database configuration in application.properties
- CRUD operations with JPA
- Custom query methods
- @Query annotation

Hands-on Project: Book API with Database Persistence

Deliverables:

- REST API connected to PostgreSQL
- Homework: Add pagination and sorting

Week 12: Building RESTful APIs with Spring Boot

Topics:

- RESTful API design best practices
- Request mapping annotations (@PostMapping, @PutMapping, @DeleteMapping)
- @PathVariable vs @RequestParam
- @RequestBody and @ResponseBody
- ResponseEntity and HTTP status codes
- Exception handling (@ExceptionHandler, @ControllerAdvice)
- Custom error responses
- API versioning

Hands-on Project: Complete RESTful API

Deliverables:

- Full CRUD API with proper REST conventions
- Homework: Add global exception handling

Week 13: Advanced Spring Boot Features

Topics:

- DTOs (Data Transfer Objects) and why they matter
- Entity to DTO conversion
- Bean validation (@Valid, @NotNull, @Size, @Email, etc.)
- Service layer architecture
- Controller-Service-Repository pattern
- Dependency injection with @Autowired and @Service
- Business logic separation
- Logging with SLF4J

Hands-on Project: Refactored Application with Proper Architecture

Deliverables:

- Layered application with validation
- Homework: Add comprehensive logging

Week 14: Final Project Development - Part 1

Goal: Design and implement core features

Activities:

- Project requirements analysis
- Database schema design
- Entity classes creation
- Repository layer implementation
- Service layer with business logic
- Basic REST endpoints

Suggested Projects:

- Task/Todo Management System
- Contact Management System
- Inventory Management System
- Student Portal System
- Simple E-commerce Backend

Deliverables:

- Complete database schema
- Core entities and repositories
- Basic CRUD functionality
- Project documentation (in-progress)

Week 15: Final Project Development - Part 2

Goal: Complete implementation and testing

Activities:

- Complete all REST endpoints
- Add validation and error handling
- Service layer completion
- Testing with Postman
- Code cleanup and refactoring
- Documentation completion
- Optional: Basic authentication

Deliverables:

- Fully functional Spring Boot application
- Complete API documentation
- Postman collection for testing
- Source code with comments
- README with setup instructions

Assessment: 35% of final grade

Week 16: Final Exam and Course Wrap-up

Activities:

Part 1: Final Exam (90 minutes) - 25% of final grade

- Section A: Multiple Choice and True/False (20 points)
  - OOP concepts, inheritance, polymorphism
  - Spring Boot annotations and architecture
  - Database and SQL concepts
  - REST API principles
- Section B: Code Reading and Debugging (30 points)
  - Identify and fix errors in provided code
  - Explain what code snippets do
  - Predict output of code
- Section C: Short Coding Problems (50 points)

    - Write a class with proper encapsulation
    - Implement inheritance hierarchy
    - Write SQL queries
    - Create a simple REST endpoint

Part 2: Project Demonstrations (30 minutes)

- Selected students present their projects
- Q&A about implementation decisions
- Peer feedback

Part 3: Course Wrap-up

- Course review and key takeaways
- Discussion of advanced topics for further learning
- Career guidance and next steps
- Resources for continued development

📊 Grading and Assessment

Grade Distribution:

Component	**Weight**	**Description**
Weekly Hands-on Exercises	5%	Completion and quality of weekly projects
Class Attendance	10%	Pay Attention during class (Q&A)
Mid-Course Project (Week 8)	10%	Library Management System
Final Project (Weeks 14-15)	25%	Spring Boot Web Application
Final Exam (Week 16)	50%	Comprehensive written exam
Total	**100%**	

Grading Scale:

Grade	**Percentage**	**Description**
A	90-100%	Excellent
B	80-89%	Good
C	70-79%	Satisfactory
D	60-69%	Needs Improvement
F	Below 60%	Fail

Assignment Submission:

- **Weekly Projects:** Submit by the end of each week

- **Mid-Course Project:** Submit by end of Week 8
- **Final Project:** Submit by end of Week 15
- **Late Submission Policy:** 10% deduction per day, maximum 3 days late

---

📝 Course Policies

Attendance and Participation:

- **Live Sessions:** Attendance is strongly encouraged but recorded sessions will be available
- **Participation:** Active participation in discussions and Q&A is valued
- **Office Hours:** Available for additional help and clarification

Academic Integrity:

- All work must be your own
- Collaboration is encouraged for learning, but submissions must be individual
- Use of AI tools (ChatGPT, etc.) for learning is allowed, but copy-pasting without understanding is prohibited
- Plagiarism will result in a zero grade and possible course removal

Communication:

- Check email and discussion forum regularly
- Ask questions during class or via forum
- Use office hours for individual help
- Response time: Instructor will respond within 24-48 hours

Technical Requirements:

- Stable internet connection for online sessions
- Working microphone and camera (camera optional but encouraged)
- Completed software setup by Week 1

🎓 Learning Outcomes by Course Sections

Section 1: Java Fundamentals (Weeks 1-5)

Students will be able to:

- Write syntactically correct Java programs
- Apply OOP principles in software design
- Implement proper exception handling
- Work with files and data persistence
- Use collections for data management

Section 2: Database Integration (Weeks 6-8)

Students will be able to:

- Design normalized database schemas
- Write efficient SQL queries
- Connect Java applications to databases using JDBC
- Implement data access layers
- Handle database transactions

Section 3: Web Development (Weeks 9-13)

Students will be able to:

- Understand REST API principles
- Build web services using Spring Boot
- Implement proper application architecture
- Use Spring Data JPA for persistence
- Apply validation and error handling

Section 4: Capstone Project (Weeks 14-16)

Students will be able to:

- Design and implement complete applications
- Make architectural decisions
- Test and debug web services
- Document code and APIs
- Deploy functional applications

🚀 Career Preparation

This course prepares students for:

- **Junior Java Developer** positions
- **Backend Developer** roles
- **Full-Stack Developer** (with frontend knowledge)
- Further studies in advanced Java topics
- Spring Boot certification preparation

Skills Gained:

- Java programming (Core and Advanced)
- Object-oriented design and analysis
- Database design and SQL
- REST API development
- Spring Boot framework
- Version control with Git
- Problem-solving and debugging
- Software architecture

📚 Additional Learning Resources

Online Platforms:

- **LeetCode:** Coding practice problems
- **HackerRank:** Java programming challenges
- **Codecademy:** Interactive Java tutorials
- **Udemy/Coursera:** Supplementary courses

Communities:

- **Stack Overflow:** Q&A for programming issues
- **Reddit:** r/learnjava, r/javahelp, r/springboot
- **GitHub:** Explore open-source Java projects
- **Discord/Slack:** Java developer communities

Official Documentation:

- **Java SE:** https://docs.oracle.com/en/java/
- **Spring:** https://spring.io/guides
- **Baeldung:** https://www.baeldung.com/ (Excellent Spring tutorials)

🆘 Student Support Services

Getting Help:

1. **During Class:** Ask questions via unmute or chat
2. **Discussion Forum:** Post questions for instructor and peer help
3. **Office Hours:** Schedule one-on-one help sessions
4. **Email:** For private concerns or questions
5. **Study Groups:** Form groups with classmates for collaborative learning

Resources for Success:

- **Recorded Sessions:** Review past classes anytime
- **Code Examples:** All class examples provided in repository
- **Sample Projects:** Reference implementations available
- **FAQ Document:** Common questions and answers
- **Quick Reference Guides:** Cheat sheets for Java, SQL, Spring Boot

---

📅 Important Dates

Week	Date	Milestone
Week 1	20 December - 26 December	Course begins
Week 8	7 February - 13 February	Mid-course project due
Week 14-15	21 March - 27 March & 28 March - 3 April	Final project development
Week 15	28 March - 3 April	Final project due

Week	Date	Milestone
Week 16	4 April - 10 April	Final exam

💡 Tips for Success

1. **Practice Daily:** Even 30 minutes of coding daily helps tremendously
2. **Start Early:** Don't wait until the deadline to start assignments
3. **Ask Questions:** No question is too basic or too silly
4. **Debug Systematically:** Read error messages carefully
5. **Use Resources:** Documentation is your friend
6. **Collaborate:** Learn from your peers
7. **Build Projects:** Apply concepts to real problems
8. **Review Regularly:** Go back to previous weeks' materials
9. **Take Breaks:** Avoid burnout, rest is important
10. **Stay Curious:** Explore beyond the syllabus

📞 Contact Information

Instructor: Set Yuth
Telegram: https://t.me/setyuth
Email: set.yuth@gmail.com
Class Hours: Saturday (6:30 pm)
Course Website: http://elearning.iic.edu.kh/course/view.php?id=1139
Discussion Forum: http://elearning.iic.edu.kh/course/view.php?id=1139
Google Meet Link: Will be provided

✅ Student Acknowledgment

By enrolling in this course, I acknowledge that I have:

- Read and understood the syllabus
- Understand the grading policy
- Agree to the academic integrity policy
- Have access to required software and tools
- Commit to completing coursework on time

Course Version: 1.0
Last Updated: December 2025
Instructor reserves the right to modify this syllabus as needed. Students will be notified of any changes.

🎯 Ready to Begin?

Welcome to Advanced Java Programming! This course will challenge you, but with dedication and practice, you'll emerge as a confident Java developer ready to build professional applications.

Let's code! 🚀