



Pilot Course Curriculum and Intervention Plan for Object-oriented programming I (BMU)

**“Improving the quality and sustainability of
learning using early intervention methods based
on learning analytics”**

Project No. 2023-1-FI01-KA220-HED-000159757



**Co-funded by
the European Union**

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Project ref. number	2023-1-FI01-KA220-HED-000159757
Project title	ISILA - Improving the quality and sustainability of learning using early intervention methods based on learning analytics
Document title	Pilot Course Curriculum and Intervention Plan for the Course: Object-oriented programming I (BMU)
Document Type	Report
Document version	1.0.0
Previous version(s)	
Planned date of delivery	January 2025
Language	English
Dissemination level	Public
Number of pages	13
Partner responsible	Belgrade Metropolitan University (BMU)
Author(s)	Jovana Jović, BMU
With contributions by:	
Revised by:	Miroslava Raspopović Milić, BMU

Table of Content

1	General course information>	4
2	Motivation and purpose (Why)	4
3	Defining more precisely what to explore (What)	5
4	Data collection strategy (How)	6
5	Data analysis and interpretation (So What)	7
6	Interventions plan (Now What)	7

1 General course information

Course name	Object-oriented programming I
Institution	Belgrade Metropolitan University (BMU)
Course level	Undergraduate
Teaching model	Hybrid – face to face teaching with online resources and assignments
Course learning objectives	<ul style="list-style-type: none"> • Mastery of basic methods of procedural programming in Java • Understanding, writing, and using methods (subprograms) in Java • Understanding and applying object-oriented programming concepts using Java • Understanding and working with arrays • Storing data in text and binary files • Ability to create simpler programs in Java

2 Motivation and purpose (Why)

Goal of the inquiry	
What do you want to learn about the teaching and learning process?	<ul style="list-style-type: none"> • Gain insights into student engagement and progress in mastering Java programming concepts and techniques, including procedural and object-oriented programming. • Improve student understanding and application of Java methods, arrays, and file handling. • Investigate the evolution of students' problem-solving and programming skills throughout the course. • Explore how students apply theoretical concepts to create simple yet functional Java programs.

3 Defining more precisely what to explore (What)

Specific questions of interest	
Key inquiry questions	<ul style="list-style-type: none"> • Which activities do students find most engaging when learning Java programming? • How does student engagement correlate with their mastery of procedural and object-oriented programming concepts in Java? • How do students progress through coding assignments, exercises, and projects in Java? • To what extent do students actively engage with course activities related to methods, arrays, and file handling? • What is the evolution of students' self-regulation and problem-solving skills as they advance through the Java programming course?
Data sources	<ul style="list-style-type: none"> • Engagement with the lectures • Exercise/assignment submission and grades • SRL weekly survey • Discord conversation data

4 Data collection strategy (How)

Data sources		<ul style="list-style-type: none"> • Learning Management System (LMS) • Quizzes and assignments • Discord student discussions • SRL surveys 	
Data aggregation		Data will be collected in xAPI format and integrated into Learning Locker either directly from the LMS plugin or through the csv2xAPI tool developed within the ISILA project	
Detailed methods for data collection			
Week#	Topic	Learning activities and materials	Data source(s) and collection method(s)
1	Introduction to Java	<ul style="list-style-type: none"> • Slides and teaching materials for the topics: <ul style="list-style-type: none"> o Specification of the Java programming language o A simple program in Java o Creating, compiling, and executing programs o Programming style and documentation o Programming errors o Developing Java programs using NetBeans/IntelliJ • Exercises for compiling and executing programs • Individual practice: Practice tasks • Test 1 	<ul style="list-style-type: none"> • Interaction with teaching materials in the course LMS • Filling out SRL survey • Assignment submissions • Assignment grades
2	Elementary Programming in Java	<ul style="list-style-type: none"> • Slides and teaching materials for the topics: <ul style="list-style-type: none"> o Reading input from the console o Identifiers and variables o Assignment statements and expressions o Naming constants, naming conventions o Numeric data types and operations o Numeric constants o JShell tool for command-line execution o Solving expressions and operator precedence 	<ul style="list-style-type: none"> • Interaction with teaching materials in the course LMS • Filling out SRL survey • Assignment submissions • Assignment grades

		<ul style="list-style-type: none"> o Case study: Determining the current time o Extended assignment operators o Increment and decrement operators o Numeric type conversions o Software development process o Case study: Counting monetary units o Common errors and pitfalls • Exercises for sample Java programs • Individual practice: Practice tasks • Test 2 	
3	Selections	<ul style="list-style-type: none"> • Slides and teaching materials for the topics: <ul style="list-style-type: none"> o Logical data types, values, and expressions o if-else statements o Common errors and pitfalls o Generating random numbers o Case study: Calculating Body Mass Index (BMI) o Case study: Tax calculation o Logical operators o Two computer science case studies o switch statements o Conditional operators o Debugging o Video demonstrations, examples, and self-assessment tests • Exercises for selections in Java programs • Individual practice: Practice tasks • Homework assignment 1 • Test 3 	<ul style="list-style-type: none"> • Interaction with teaching materials in the course LMS • Filling out SRL survey • Assignment submissions • Assignment grades
4	Mathematical Functions, Symbols, and Strings	<ul style="list-style-type: none"> • Slides and teaching materials for the topics: <ul style="list-style-type: none"> o Common mathematical functions o char data type for symbols and operations o String class o Case study: Guessing a birthday o Case study: Converting a hexadecimal digit o Case study: Revising the lottery program o Formatting console output o Video demonstrations, examples, and self-assessment tests • Exercises for mathematical, functions, symbols and strings in Java programs 	<ul style="list-style-type: none"> • Interaction with teaching materials in the course LMS • Filling out SRL survey • Assignment submissions • Assignment grades

		<ul style="list-style-type: none"> ● Individual practice: Practice tasks ● Test 4 	
5	Loops	<ul style="list-style-type: none"> ● Slides and teaching materials for the topics: <ul style="list-style-type: none"> ○ Introduction to loops ○ while loop ○ Case study: Guessing numbers ○ Strategies for loop design ○ Controlling loops by limiting values ○ do-while loop ○ for loop ○ Choosing the right loop ○ Nested loops ○ Minimizing numerical errors ○ Keywords: break and continue ○ Case studies ○ Video demonstrations, examples, and self-assessment tests ● Exercises for loops in Java programs ● Individual practice: Practice tasks ● Test 5 	<ul style="list-style-type: none"> ● Interaction with teaching materials in the course LMS ● Filling out SRL survey ● Assignment submissions ● Assignment grades
6	Methods	<ul style="list-style-type: none"> ● Slides and teaching materials for the topics: <ul style="list-style-type: none"> ○ Defining methods ○ Calling methods ○ void vs. methods returning values ○ Passing arguments by value ○ Modular programming ○ Case study: Hexadecimal-to-decimal conversion ○ Method overloading ○ Variable scope ○ Case study: Generating random characters ○ Method abstraction and incremental refinement ○ Implementation details ○ Video demonstrations, examples, and self-assessment tests ● Exercises for methods in Java programs ● Individual practice: Practice tasks ● Homework assignment 2 ● Test 6 	<ul style="list-style-type: none"> ● Interaction with teaching materials in the course LMS ● Filling out SRL survey ● Assignment submissions ● Assignment grades
7	Single-Dimensional Arrays	<ul style="list-style-type: none"> ● Slides and teaching materials for the topics: <ul style="list-style-type: none"> ○ Basics of arrays ○ Case study: Analyzing numbers ○ Case study: Deck of cards ○ Copying arrays 	<ul style="list-style-type: none"> ● Interaction with teaching materials in the course LMS ● Filling out SRL survey ● Assignment submissions ● Assignment grades ● Discord student discussion

		<ul style="list-style-type: none"> o Passing arrays to methods o Case study: Counting letter occurrences o Variable-length argument lists o Searching arrays o Sorting arrays o Array class o Command-line arguments o Video demonstrations, examples, and self-assessment tests ● Exercises for single-dimension array in Java programs ● Individual practice: Practice tasks ● Test 7 ● Essay 1 ● Discord discussion essay 1 	
8	Multi-Dimensional Arrays	<ul style="list-style-type: none"> ● Slides and teaching materials for the topics: <ul style="list-style-type: none"> o Basics of multi-dimensional arrays o Processing two-dimensional arrays o Passing two-dimensional arrays to methods o Case study: Grading a multiple-choice test o Case study: Finding the closest pair o Case study: Sudoku o Multi-dimensional arrays o Video demonstrations, examples, and self-assessment tests ● Exercises for multi-dimension array in Java programs ● Individual practice: Practice tasks ● Test 8 	<ul style="list-style-type: none"> ● Interaction with teaching materials in the course LMS ● Filling out SRL survey ● Assignment submissions ● Assignment grades
9	Objects and Classes	<ul style="list-style-type: none"> ● Slides and teaching materials for the topics: <ul style="list-style-type: none"> o Defining classes for objects o Constructing objects using constructors o Accessing objects through reference variables o Using classes from the Java library o Static variables, constants, and methods o Visibility modifiers o Encapsulation of data fields o Passing objects to methods o Arrays of objects o Immutable objects and classes, and their scope o this reference 	<ul style="list-style-type: none"> ● Interaction with teaching materials in the course LMS ● Filling out SRL survey ● Assignment submissions ● Assignment grades

		<ul style="list-style-type: none"> o Video demonstrations, examples, and self-assessment tests • Exercises for object and classes in Java programs • Individual practice: Practice tasks • Homework assignment 3 • Test 9 	
10	Object-Oriented Thinking	<ul style="list-style-type: none"> • Slides and teaching materials for the topics: <ul style="list-style-type: none"> o Abstraction and encapsulation of classes o Thinking in objects o Class relationships o Case study: Designing the Course class o Case study: Designing a class for stacks o Handling primitive data types as objects o Automatic type conversion o Classes BigInteger and BigDecimal o String class o StringBuilder and StringBuffer classes o Video demonstrations, examples, and self-assessment tests • Exercises for object oriented thinking in Java programs • Individual practice: Practice tasks • Test 10 	<ul style="list-style-type: none"> • Interaction with teaching materials in the course LMS • Filling out SRL survey • Assignment submissions • Assignment grades
11	Inheritance and Polymorphism	<ul style="list-style-type: none"> • Slides and teaching materials for the topics: <ul style="list-style-type: none"> o Superclasses and subclasses o Using the super keyword o Method overriding o Polymorphism o Dynamic binding o Object conversion and the instanceof operator o The equal() method for objects o ArrayList class o Useful methods for lists o Case study: Custom stack class o Protected data and methods o Preventing extension and overriding o Video demonstrations, examples, and self-assessment tests • Exercises for inheritance and polymorphism in Java programs • Individual practice: Practice tasks • Test 11 	<ul style="list-style-type: none"> • Interaction with teaching materials in the course LMS • Filling out SRL survey • Assignment submissions • Assignment grades

<p>12</p>	<p>Working with Exceptions and Input/Output of Textual Data</p>	<ul style="list-style-type: none"> ● Slides and teaching materials for the topics: <ul style="list-style-type: none"> ○ Working with exceptions ○ Types of exceptions ○ Throwing, declaring, and catching exceptions ○ Using exceptions ○ Defining custom exception classes ○ File class ○ Files for reading and writing textual data ○ Reading data from the web ○ Case study: Web scraper ○ Video demonstrations, examples, and self-assessment tests ● Exercises for exceptions in Java programs ● Individual practice: Practice tasks ● Homework assignment 4 ● Test 12 	<ul style="list-style-type: none"> ● Interaction with teaching materials in the course LMS ● Filling out SRL survey ● Assignment submissions ● Assignment grades
<p>13</p>	<p>Abstract Classes and Interfaces</p>	<ul style="list-style-type: none"> ● Slides and teaching materials for the topics: <ul style="list-style-type: none"> ○ Abstract classes ○ Case study: Abstract number class ○ Case study: Calendar and Gregorian calendar ○ Interfaces ○ Comparable interface ○ Clonable interface ○ Interfaces vs. abstract classes ○ Case study: Rational class ○ Guidelines for class design ○ Video demonstrations, examples, and self-assessment tests ● Exercises for exceptions in Java programs ● Individual practice: Practice tasks ● Test 13 	<ul style="list-style-type: none"> ● Interaction with teaching materials in the course LMS ● Filling out SRL survey ● Assignment submissions ● Assignment grades
<p>14</p>	<p>Input and Output of Binary Data</p>	<ul style="list-style-type: none"> ● Slides and teaching materials for the topics: <ul style="list-style-type: none"> ○ How to manage text I/O in Java? ○ Text I/O vs. binary I/O ○ Binary I/O classes ○ Case study: Copying files ○ Object input and output ○ Files with random access ○ Video demonstrations, examples, and self-assessment tests ● Exercises for exceptions in Java programs ● Individual practice: Practice tasks 	<ul style="list-style-type: none"> ● Interaction with teaching materials in the course LMS ● Filling out SRL survey ● Assignment submissions ● Assignment grades

		<ul style="list-style-type: none"> • Test 14 	
15	Testing with JUnit	<ul style="list-style-type: none"> • Slides and teaching materials for the topics: <ul style="list-style-type: none"> o Basics of JUnit o Using JUnit in NetBeans o Test programs • Exercises for exceptions in Java programs • Individual practice: Practice tasks • Homework assignment 5 • Test 15 • Course project assignment • Project • Discord discussion about the project • Discord discussion about the essay 2 	<ul style="list-style-type: none"> • Interaction with teaching materials in the course LMS • Filling out SRL survey • Assignment submissions • Assignment grades • Project grades • Discord student discussion

5 Data analysis and interpretation (So What)

<p>Sense making and interpretation context</p>	<p>Use dashboards to visualize student engagement levels in Java programming activities.</p> <p>Analyze correlations between course activities (e.g., coding exercises, case studies, and tests) and student performance outcomes.</p> <p>Identify students in the bottom quartile based on activity completion, self-regulation, and programming skills.</p> <p>Compare student results with established course goals (e.g., mastering Java concepts) and prior performance expectations.</p>
---	--

6 Interventions plan (Now What)

<p>Potential interventions</p>	<p>Face-to-face interventions:</p> <ul style="list-style-type: none"> - devote more time to the topic/assignment, - arrange additional learning activities, - address challenges, - extend deadlines if students are struggling. <p>Internet-based interventions:</p> <ul style="list-style-type: none"> - email class-wide reminders, - recommendations, - personalized email offering 1-to-1 tutorial or additional support, - share additional resources on Discord. <p>Content adjustments:</p> <ul style="list-style-type: none"> - revise content or format for topics with low engagement (e.g., simplify lecture slides, add examples).
---------------------------------------	--