

תכנות מונחה עצמים

Introduction to Object Oriented Programming

89-111

Course type: lecture+recitation

סוג הקורס: הרצאה + תרגול

Extent: 3+2 hours per week

היקף שעות שבועיות: 2+3

Website:

<https://lemida.biu.ac.il>

אתר הקורס באינטרנט:

הקורס יועבר בשפה העברית או האנגלית, לבחירת המרצה.

The course will be taught in Hebrew or English, depending on the lecturer's decision.

א. מטרת הקורס:

הקניית ידע בתכנות ועיצוב קוד מונחה עצמים. התנסות בשפת התכנות ג'אווה.

Outcomes: Fundamentals of programming and design of object-oriented code. Practicing the Java programming language.

ב. תוכן הקורס:

להלן רשימת הנושאים שידונו בקורס, לא בהכרח בסדר הזה:

The course will introduce the following topics, not necessarily in this order:

Administration, introduction to object oriented programming (OOP). Java vs. C and python Recitation: introduction to the work environment and exercise submission	מנהלות, הקדמה לתכנות מונחה עצמים. ג'אווה לעומת C, ג'אווה לעומת פייתון תרגול: היכרות עם סביבת העבודה והגשת תרגילים	1
Classes and objects, constructors, reference types and the memory in java, public/private access modifiers, static methods and fields, overloading and scoping Recitation: writing a class, useful java types	מחלקות ועצמים, בנאים, טיפוסי הפניה ותמונת הזיכרון בג'אווה, נראות פומבית ופרטית, מתודות ושדות סטטיים, העמסה ו-scoping תרגול: כתיבת מחלקה, היכרות עם טיפוסים שימושיים	2
Code testing, test-driven development (TDD), packages and documentation	בדיקות קוד, תכנות מונחה בדיקות, עוד דוגמאות קוד מונחה עצמים, חבילות ותיעוד קוד	3

Recitation: introduction to Exception handling, code testing tools	תרגול: הקדמה לטיפול בשגיאות, שימוש בכלי בדיקות	
Interfaces, polymorphism, polymorphic types, contracts and invariants Recitation: writing and using interfaces	ממשקים, פולימורפיזם, טיפוסים נתונים פולימורפיים, חוזים ומשתמרים תרגול: כתיבה ומימוש ממשקים, היכרות בסיסית עם אוספים	4
Programming principles: encapsulation, abstraction, delegation, java generics, introduction to java collections Recitation: encapsulation examples, Factory design pattern, refactoring	עקרונות תכנות: הכמסה, הפשטה, האצלה, טיפוסים גנריים בג'אווה, מבוא לאוספים בג'אווה תרגול: דוגמאות להכמסה, תבנית עיצוב מפעל (הפרדה בין ממשק למימוש), refactoring	5
Type hierarchy. UML diagram representation, extending interfaces, extending classes, abstract types, overriding methods Recitation: inheritance, abstract, overriding, @Override, Object base class	היררכיות טיפוסים. ייצוג בעזרת תרשים UML, הרחבת ממשקים, הרחבת מחלקות, טיפוסים מופשטים, דריסת מימושים. תרגול: חזרה על הורשה, הפשטה, דריסה, @Override, מחלקת הבסיס Object	6
I/O and exceptions. File system, reading bytes, text and encoding, streams, nio and buffering. Exception hierarchy in java, checked/unchecked, principles of throwing and handling exceptions. Recitation: serial versus random access, synchronous and asynchronous IO. Handling different types of errors.	קלט\פלט וחריגים. מערכת הקבצים, קריאת בתים, טקסט וקידוד, זרמים, nio ו-buffering. היררכיית החריגים בג'אווה, checked/unchecked, עקרונות נכונים של שימוש בזריקת שגיאות ותפיסת שגיאות. תרגול: גישה סדרתית לעומת גישה רנדומית, IO סינכרוני וא-סינכרוני. הדגמת טיפול בשגיאות במצבים שונים.	7
Object serialization including the java-provided solution and building an independent solution, linked data types, schema. Network and Sockets Recitation: sending objects over the network / communication between processes	סידורת של עצמים כולל הפתרון המובנה בג'אווה ובניית פתרון עצמאי, מבנים מקושרים, סכמה ועוד, רשת, Sockets תרגול: שליחת עצמים ברשת \ תקשורת בין תהליכים	8

<p>Design patterns. Template method, observer, abstract factory, SOLID and GRASP principles, classification of OOP design patterns</p> <p>Recitation: More design patterns (Composite), more on IO, comparator, comparable, regexps</p>	<p>תבניות עיצוב. Template Method, abstract factory, observer, עקרונות עיצוב כלליים (GRASP, SOLID) סיווג כללי של תבניות מרכזיות בתכנות מונחה עצמים</p> <p>תרגול: תבניות עיצוב נוספות (composite) המשך, IO, comparator, comparable, regexps</p>	9
<p>Inheritance issues: Decorator design pattern, Builder design patter, Liskov substitution principle, tight type coupling in inheritance, writing generic types, nested classes</p> <p>Recitation: nested classes, iterators</p>	<p>סוגיות בהורשה: תבנית עיצוב קשטן (decorator) לצורך החלפת הורשה בהאצלה, תבנית עיצוב builder, עקרון ההחלפה, צמידות טיפוסים בהורשה, כתיבת טיפוסים גנריים, מחלקות מקוננות</p> <p>תרגול: מחלקות מקוננות, איטרטורים</p>	10
<p>Immutable types. Writing immutable types, producer queries, builders, copying and deep cloning.</p> <p>Recitation: copying methods, prototype design pattern</p>	<p>טיפוסים מקובעים. כתיבת טיפוסים מקובעים, שאילתות מפיקות, שימוש ב-builders, העתקה ושיבוט עמוק.</p> <p>תרגול: מתודות העתקה, prototype design pattern</p>	11
<p>Advanced inheritance topics (static versus dynamic type), dynamic dispatch and static binding, overloading, constructors and object initialization, equals and hashCode</p> <p>Recitation: Enums, overriding rehearsal</p>	<p>נושאים מתקדמים בהורשה (טיפוס סטטי ודינאמי, הקצאה סטטית ודינאמית, העמסה, בנאים ואתחול עצמים, מתודות hashCode – equals</p> <p>תרגול: Enums, חזרה על מנגנון הדריסה</p>	12
<p>OOP in python. Basic syntax, multiple inheritance, dynamic inheritance, properties, abstract classes, enforcement by programmers versus compiler enforcement, class methods</p> <p>Recitation: python operators and iterators, rehearsal.</p>	<p>תכנות מונחה עצמים בפייתון. היכרות עם התחביר הבסיסי, הורשה מרובה, הורשה דינאמית, properties, אכיפה ע"י המתכנתים לעומת אכיפה ע"י הקומפיילר, מתודות מחלקה</p> <p>תרגול: דוגמאות נוספות בפייתון, אופרטורים ואיטרטורים, חזרה למבחן.</p>	13-14

ג. חובות הקורס:

דרישות קדם: הקורס 89-110

חובות / דרישות / מטלות: מבחן + תרגילים מעשיים

מרכיבי הציון הסופי (ציון מספרי / ציון עובר): 80% מבחן + 20% תרגול. כדי לעבור את הקורס יש לקבל ציון עובר הן ברכיב התרגילים והן ברכיב המבחן

Requirements:

Prerequisites: course 89-110

Assignments and other duties: final exam + programming exercises

Grade: 80% final exam, 20% assignments. A passing grade in both is required to pass the course.

ד. ביבליוגרפיה:

Bibliography:

Recommended reading:

The Java™ Tutorials <https://docs.oracle.com/javase/tutorial/>,

Specifically <https://docs.oracle.com/javase/tutorial/java/index.html>

Design Patterns: Elements of Reusable Object-Oriented Software

by Erich Gamma, John Vlissides, Ralph Johnson, and Richard Helm

Java in a Nutshell: A Desktop Quick Reference

by Ben Evans and David Flanagan, 2018.

Building Java Programs: A Back to Basics Approach, 5th edition

by Stuart Reges and Marty Stepp, 2019