

קורס מתקדם לפיתוח חשיבה Object Oriented

Design Patterns-I

מטרת הקורס:

להקנות חשיבה object oriented מקצועית ומתקדמת למפתחים בשפות OO, תוך היכרות עם ה- design patterns העיקריים, עם אוריינטציה חזקה לעולם התעשייה האמיתי.

פירוט:

רוב שפות התכנות בתעשייה היום הן שפות Object Oriented. כדי לנצל נכון את יכולות השפות האלו, יש צורך לפתח חשיבה וצורת מבט מתאימים, אחרת היתרונות הגדולים של השפה לא יבואו לידי ביטוי.

בקורס זה נרכוש כלים לפיתוח מתקדם של מערכות object oriented, בשילוב עם דוגמאות רלוונטיות מעולם התעשייה. נלמד את עקרונות ה-SOLID, נעבור על ה-Design Patterns העיקריים ונתרגל design של מודולים לפיתוח. הדגש החזק הוא על פיתוח חשיבה מונחת עצמים, ושימוש בתבניות העיצוב ככלי להשגת מטרה זו.

הקורס מועבר על ידי ד"ר מירי (קופל) בן-ניסן, מרצה בכירה באוניברסיטה מזה כ-20 שנה למקצועות תכנות, ובעלת ניסיון עשיר ורב בעולם הפיתוח כמפתחת, ארכיטקטית ומנהלת צוותי פיתוח בעולמות האלגוריתמים, mobile-I RT.

משך הקורס: כ-40 שעות.

קהל היעד: מפתחים בעלי ידע וניסיון של שנה לפחות באחת משפות ה-Object Oriented. יש אפשרות להאריך את הקורס על מנת להכשיר מפתחים בלי ניסיון בשפה או כחלק מהסבה לפיתוח.

מתודיקה: הקורס יורכב מחלקים פרונטליים ומתרגולים קצרים. למשתתפים ימסרו כל החומרים כולל השקפים וקטעי הקוד.

עלות: 24000 ש"ח + מע"מ.

סילבוס מקוצר: ראשי פרקים של הנושאים שיילמדו בקורס:

- הגדרות של שפה Object Oriented.
- Prefer Composition over Inheritance – מוטיבציה ודוגמאות.
- עקרונות SOLID לפיתוח מונחה עצמים.
- UML – class diagrams and sequence diagrams.
- מימוש של design patterns נבחרים. יתרונות וחסרונות. מתי להשתמש ומתי לא.
- Anti-patterns.

סילבוס מפורט בעמוד הבא.

קורסי המשך: קורסת תכנות בשפת C++ למתכנתים.

קורס תכנות C++ מתקדם – (11/14/17) modern C++.

לפרטים: ד"ר מירי (קופל) בן-ניסן
050-6349742
miri@mkbn.co.il

סילבוס מפורט:

- **OOP Concepts**
 - Object Oriented Software development life cycle
 - OO Design Concepts and considerations
 - Prefer composition over inheritance – motivation & examples
 - **SOLID** concepts – examples and best practices
- **UML best practices**
 - Structural Diagrams, Behavioral Diagrams, Interaction Diagrams.
- **Design Patterns**
 - **Creational Patterns**
 - Factory Patterns, Prototype.
 - Singleton – the REAL motivation, thread-safe singleton.
 - Builder – motivation, implementation and hands-on exercise.
 - **Behavioral Patterns**
 - Visitor – understand deeply the OOP concepts. Hands-on exercise.
 - Iterator – the OCP and the pattern, hands-on exercise building an iterator.
 - Strategy – several implementations.
 - Observer, Template method, Interpreter.
 - Command – hands-on exercise: build a thread-pool system.
 - Mediator, Chain of responsibility.
 - **Structural Patterns**
 - Flyweight, Composite, Decorator.
 - Façade, Proxy, Adapter.
 - Bridge.
- **Architectural Patterns – Overview of main patterns**
 - Layered architecture
 - Event driven
 - Microservices architecture
 - Domain driven design (DDD) architecture

תאריך עדכון: 25.9.2019