

המחלקה להנדסת תוכנה  
הנדסת תוכנה (קורס מס' 10014)  
סמסטר ב' תשע"ה 2014/14  
סילבוס הקורס

**מתכונת הקורס:**

מרצה ומעבדה: ד"ר ראובן יגל, משרד 106, שעות קבלה לאחר ההרצאה בתיאום, מייל [robi@post.jce.ac.il](mailto:robi@post.jce.ac.il), גיטהאב: robi-y  
מתרגל: יורי שמורגון, [yuri.jce2015@gmail.com](mailto:yuri.jce2015@gmail.com), גיטהאב: MeLight  
הגשות ובדיקה: טופס הגשות \ מייל המתרגל \ פורום - לפי הנחיות הרצאה: 3 שעות, מעבדה: שעה אחת, תרגיל: שעה אחת, סה"כ 4 נ"ז שעות ומיקום – ראו במערכת ידיעון  
אתר הקורס: <https://github.com/jce-il/se-class/wiki>  
דרכי התקשרות - עדיפות הפנייה בסדר יורד:  
- השתתפות ושאלות בהרצאה ובתרגיל ☺  
- קריאת חומרי הקורס וההגדרות למשימות השונות, התייעצות עם חברים, חיפוש,  
דיון בפורום הקורס – <https://gitter.im/jce-il/se-class>  
- מייל  
- שעות קבלה  
- עדיף שלא לפנות דרך מערכת הידיעון ☹

**מטרות הקורס:**

הבנת הבעיות והפתרונות המרכזיים של הנדסת תוכנה לפיתוח מוצרי תוכנה גדולים. פיתוח ראייה מערכתית והנדסית. היכרות עם תהליכים, שיטות עבודה, כלים מעודכנים בשימוש התעשייה ויכולת בחירה ביניהם. לימוד והתנסות בבניית פרויקט תוכנה משמעותי תוך כדי עבודת צוות וכישורים רבים נוספים. הכנה לפרויקט הגמר.

**מבנה הקורס:**

פרויקט קבוצתי – לפי שלבים שיוצגו במהלך הקורס, סקרים ופגישות תכנון עבור הפרויקט (חלקם בזמן ההרצאה, המעבדה או התרגילים, חובת נוכחות לנציגי הצוות המציג)  
חומר תיאורטי ויישומי – דיון, הדגמות, הכנה לשלבי הפרויקט השונים  
משימות אישיות – תרגילים קצרים ולימוד עצמי של נושאים נדרשים (אב-טיפוס, git, TDD, Legacy)  
תרגיל – לימוד ועזרה בנושאים טכניים, השלמת סקרים ומשימות התקדמות בפרויקט מבחן מסכם

**תכנית הקורס (כפוף לשינויים, לו"ז מעודכן באתר הקורס):**

- מבוא
  - מוטיבציה להנדסת תוכנה
  - תהליכים מודלים ותקנים לפיתוח תוכנה
  - פתוח תוכנה אדפטיבי - Agile
- תקשורת ותכנון
  - התנעת וניהול פרויקט
  - עבודת צוות
  - עבודה עם לקוח, כתיבת מפרטים
  - הערכות וניתוח סיכונים, סקרים, משוברים ותכנונים
- ניתוח/מידול/ליכון/בדיקות
  - ניתוח דרישות (SRS, User stories, Use cases)

- תיכון (מונחה עצמים), ארכיטקטורה, ממשקים, UML, SDS
- מימוש
  - בדיקות ופיתוח מונחה בדיקות
  - עקרונות קידוד מונחה עצמים, תבניות תיכון (Design Patterns)
  - חווית משתמש, תחזוקה
- כלים ושיטות
  - כלי CASE, ניהול תצורה - בקרת קוד (VCS), בדיקות, סקרי קוד, שילוב והפצה מתמשכים (CI/CD), שחלוף - Refactoring ועוד
- מרצה אורח מהתעשייה (בד"כ)
- נושאים נוספים: תיעוד, זכויות יוצרים, קוד פתוח ועוד

### קורסים קשורים:

קורסי תכנות, תיכון מונחה עצמים, מפרטים פורמאליים לתיאור מערכות, פרויקט גמר ועוד (ראו ידיעון)

### חובות וצינונים:

- פרויקט - 60%, עבודת צוות, הציון מורכב משלבים שונים שיימסרו במהלך הקורס באתר, כגון: הצעת פרויקט, מפרט דרישות, מפרט תיכון וסבבי פיתוח. בכל שלב יפורטו התאריכים לביצוע (הגשה בד"כ בחצות לפני ההרצאה הבאה ועוד שבוע למקצה שיפורים) ומרכיבי הציון השונים כגון: איכות המימוש, משוברים וסקרים ועבודה עם לקוחות. הציון הוא קבוצתי אך יתכנו תיקונים משמעותיים לפי הערכת עמיתים (חובה למלא בסיום כל סבב לכל הצוות כולל המעריך), תרומה לפרויקט והתרשמות צוות הקורס.
- ציון אישי - 15%, תרגילים אישיים (הסבר בד"כ מופיע בסוף מצגת ההרצאה, הגשה עד להרצאה הבאה)
  0. רישום לקורס (ללא ציון)
  1. הכרת סביבת הפיתוח וההפצה
  2. משוב למסמך תיכון
  3. בקרת תצורה
  4. פיתוח מונחה בדיקות
  5. עבודה עם קוד קיים – בונוס
  6. [הקלדה עיוורת](#)
  7. הצגה סופית
- מבחן – 25% - על חומרי ההרצאה, חובת מעבר
- בונוס – עד 5 נקודות תוספת על השתתפות ותרומה משמעותיים לקורס (במיוחד בהרצאה ובפורום).

איחורים: במקרה של סיבה מוצדקת בלבד יש לתאם מראש לו"ז חלופי, תיתכן הורדת ניקוד בהתאם לנסיבות ולזמן. יש לגבות כל סיכום אחר עם צוות הקורס במייל עם אישור. יושרה: שימוש לא ראוי בחומרים של אחרים יטופל בהתאם. אין להפעיל קבלני משנה בפרויקט ללא אישור מצוות הקורס.

### חומרי עזר:

אתר הקורס מכיל קישורים לחומרים השונים ובפרט במצגות ההרצאה. בנוסף מצורפת רשימת ספרות להרחבה:

#### General/Classical Course Textbooks (not mandatory):

- Pressman R. S., "Software Engineering – A Practitioner's Approach", 8e, McGraw-Hill, 2014.

- Sommerville I., "Software Engineering", 9e, Addison Wesley, 2010.
- Schach S. "Object-Oriented Software Engineering", 8e, McGraw-Hill, 2010.
- Laplante P. A., "What Every Engineer Should Know about Software Engineering", CRC, 2007.
- Jalote P., "A Concise Introduction to Software Engineering", Springer, 2008.

#### Software Engineering:

- Beck K., "Test Driven Development: By Example", Addison-Wesley, 2002
- McConnell S., "Code Complete: A Practical Handbook of Software Construction", 2e, Microsoft Press, 2004 (Hebrew Version, 2013).
- McConnell S., "Rapid Development: Taming Wild Software Schedules", Microsoft Press, 1996.
- McConnell S., "Software Project Survival Guide", Microsoft Press, 1997.
- Hunt A. and Thomas D., "The Pragmatic Programmer: From Journeyman to Master", Addison-Wesley, 1999.
- Martin R. C., "Agile Software Development, Principles, Patterns, and Practices", Prentice Hall, 2002 (also a C# version).
- Kniberg H., "Scrum and XP from the Trenches", Lulu.com, 2007.
- Sink E., "Version Control by Example", 2011.
- Metz S., Practical Object-Oriented Design in Ruby an Agile Primer, Addison-Wesley Professional, 2012.

#### Background:

- Fowler M., "UML Distilled: A Brief Guide to the Standard Object Modeling Language", 3e, Addison-Wesley, 2003.
- Berkun S., Making Things Happen: Mastering Project Management 2<sup>nd</sup> ed., Microsoft Press, 2010.
- Gamma E. et. al., "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, 1994.
- DeMarco T. and Lister T., "Peopleware", 2e, Dorset House, 1999.
- Brooks F. P., "The Mythical Man-Month: Essays on Software Engineering", 2e, Addison-Wesley, 1995.

Programming: Python, Web, GAE, etc. see wiki & presentations

בהנאה ובהצלחה