

המחלקה להנדסת תוכנה
קורס הנדסת תוכנה 10014
סמסטר א' תשע"ז 2016/17
סילבוס הקורס

מתכונת הקורס:

מרצה: ד"ר ראובן יגל, משרד 106, שעות קבלה לפניאחרי ההרצאה בתיאום, מייל robi@post.jce.ac.il, גיטהאב: [robi-y](https://github.com/robi-y)
מתרגלת מח"ר: הדר פיקאלי, hadarp,hadarp10@gmail.com
הגשת משימות: טופס הגשות \ מייל המתרגל \ פורום \ github issue - לפי הנחיות שיפורסמו
הרצאה: 3 שעות, תרגיל: 2 שעות סה"כ 4 נ"ז.
מומלץ להירשם בנפרד לקורס בחירה הנדסת תוכנה בשירות הקהילה—10015.
שעות ומיקום – ראו במערכת ידיעון
אתר הקורס: <https://github.com/jce-il/se-class/wiki>
אתר חומרי הקורס: <http://jce-il.github.io/se-class17a/>
פורום טכני (צ'אט) gitter.im/.../2017a, בנושאי ציונים פניות במייל בלבד.

מטרות הקורס:

הבנת הבעיות והפתרונות המרכזיים של הנדסת תוכנה לפיתוח מערכות תוכנה בהיקף גדול. פיתוח ראייה מערכתית והנדסית. היכרות עם תהליכים, מיומנויות, שיטות עבודה, כלים מעודכנים בשימוש התעשייה ויכולת בחירה ביניהם. לימוד והתנסות בבניית פרויקט תוכנה משמעותי תוך כדי עבודת צוות וכישורים רכים 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). הפרויקט יתבצע עבור לקוח (חיצוני) בקבוצות של כחמישה סטודנטים.
הרצאה - חומר תיאורטי ויישומי – דיון, הדגמות, הכנה לשלבי הפרויקט השונים. מומלץ להביא מחשב נייד.
משימות אישיות – תרגילים קצרים ולימוד עצמי של נושאים נדרשים.
תרגיל ומעבדה – לימוד טכנולוגיות ועזרה בנושאים טכניים, השלמת סקרים ומשימות התקדמות בפרויקט.
מבחן מסכם.

תכני הקורס (כפוף לשינויים, לו"ז ומצגות מעודכנים באתר חומרי 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), טיוב קוד - Refactoring ועוד
- מרצה אורח מהתעשייה (בד"כ)
- נושאים נוספים: תיעוד, קוד פתוח ועוד

קורסים קשורים:

קורסי תכנות, תכנות בסביבת אינטרנט, תיכון מונחה עצמים, ניהול פרויקטי תוכנה, פרויקט גמר ועוד (ראו ידיעון)

חובות וצינונים:

- הסטודנט נדרש להירשם למערכות הקורס המקוונות ולוודא שמעודכן בתדירות בפורום. מומלץ להביא מחשב נייד להרצאה – יינתן זמן לתרגול ועבודה על הפרויקט בהקשר לחומרי ההרצאה.
 - 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) ומרכיבי הציון השונים כגון: רלוונטיות ואיכות 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, משוב מהלקוח והתרשמות צוות הקורס.
 - ציון אישי - 15%, תרגילים אישיים (הסבר בד"כ מופיע בסוף מצגת ההרצאה או במאגר הרלוונטי והגשה עד להרצאה הבאה), כוללים: הכרת סביבת הפיתוח (תיעוד, בקרת תצורה והפצה), פיתוח אפליקציית רשת, בקרת תצורה, פיתוח מונחה בדיקות, עבודה עם קוד קיים, הקלדה עיוורת.
 - מבחן - 20%, בעיקר על חומרי ההרצאה, חובת מעבר, תיתכן אפשרות תרומת שאלות וכן פטור ע"י סיכום כיתתי.
 - תרומה לקורס - 5%, השתתפות פעילה בהרצאה ובפורום, הגשת תרגילי כיתה? במהלך הסמסטר יפורסמו ציוני ביניים (לא סופיים) ומשובים בהקדם האפשרי.
- איחורים: במקרה של סיבה מוצדקת בלבד יש לתאם מראש לו"ז חלופי עם המרצה בלבד במייל, תיתכן הורדת ניקוד בהתאם לנסיבות ולזמן.
- יושרה: אסור שימוש לא ראוי בחומרים שיוצרו של אחרים. 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", 10e, Addison Wesley, 2015.
- 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.
- Visser J., Building Maintainable Software - Ten Guidelines for Future-Proof Code, O'Reilly Media, 2016.

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 2nd 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: Web, Cloud, etc. see course materials

בהנאה ובהצלחה