

המחלקה להנדסת תוכנה
קורס הנדסת תוכנה 10014
סמסטר ב' תשע"ח 2018
סילבוס הקורס

מתכונת הקורס:

מרצה: ד"ר ראובן יגל, משרד D114, שעות קבלה אחרי ההרצאה בתיאום, מייל robi@post.jce.ac.il, גיטהאב: [robi-y](https://github.com/robi-y)
מתרגל ובודק: ערן חדד, eran.hadd@gmail.com, גיטהאב: [eranhhd](https://github.com/eranhhd)
הגשת משימות: **עדכון מאגר** \ 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 (ראו סילבוס נפרד).
שעות ומיקום – ראו במערכת ידיעון
ויקי הקורס: <https://github.com/jce-il/se-class/wiki>
חומרי הקורס: <https://github.com/jce-il/se-class-materials> - ולא בתחנת המידע
פורום טכני (צ'אט) gitter.im/.../se-class18b (ציבורי!), פניות בנושאי איחורים וציונים במייל בלבד.

מטרות הקורס:

הבנת הבעיות והפתרונות המרכזיים של הנדסת תוכנה לפיתוח מערכות ומוצרי תוכנה בהיקף גדול. פיתוח ראייה מערכתית והנדסית. היכרות עם תהליכים, מיומנויות, שיטות עבודה וכלים מעודכנים בשימוש התעשייה עם יכולת בחירה ביניהם. לימוד והתנסות בבניית פרויקט תוכנה משמעותי תוך כדי עבודת צוות וכישורים רכים נוספים. הכנה לפרויקט הגמר.

מבנה הקורס:

פרויקט קבוצתי – מפותח בסבבים ולפי שלבים שיוצגו במהלך הקורס, סקרים ופגישות תכנון עבור הפרויקט (חלקם בזמן ההרצאה, המעבדה או התרגילים, חובת נוכחות לנציגי הצוות המציג). הפרויקט יתבצע **בעדיפות** עבור לקוח (חיצוני) בקבוצות של ארבע עד חמישה סטודנטים.
הרצאה - חומר תיאורטי ויישומי – דיון, הדגמות, הכנה לשלבי הפרויקט השונים ועזרה במשימות אישיות.
משימות אישיות – תרגילים קצרים ולימוד עצמי של נושאים נדרשים.
תרגיל ומעבדה – לימוד טכנולוגיות ועזרה בנושאים טכניים, השלמת סקרים ומשימות התקדמות בפרויקט.
מבחן מסכם.

תכנית שבועית (כפוף לשינויים, לו"ז ומצגות מעודכנים באתר חומרי הקורס):

הרצאה	תרגיל	
1 מבוא להנדסת תוכנה לוגיסטיקה – סילבוס \ אתר הקורס <u>הנדסת תוכנה בשירות הקהילה – מבוא ויריד</u> הפרויקט – מטרות והדגמה משימת פרויקט 1: <u>בחירת רעיון</u>	משימה אישית 1 – רישום למערכות הקורס, הכרות עם סביבת העבודה בקורס, הפצת אפליקציית Angular 2 ראשונה	
2 עבודת צוות, <u>חשיבה עיצובית</u> השלמת לוגיסטיקה: מבחן וגליון ציונים משימת פרויקט 2: הקמת צוות, אתחול הפרויקט והמאגר	פיתוח צד לקוח – Angular. היכרות עם Component ותקשורת בין Components. כתיבת Service ועבודה נכונה עם Angular 2	
3 דרישות - מפרט דרישות תוכנה (SRS), תרחישי שימוש, כלי מידול UML, סיפורי	המשך בניית צד לקוח (עבור אב טיפוס, Wireframing, routing)	

היכרות עם <i>Material Design</i> והטמעה בקוד	משתמש, מפרטים מורצים, סדנת דרישות סקר אתחול פרויקט משימת פרויקט 3 – כתיבת מפרט דרישות (כולל אב טיפוס) בתרגיל	
השלמת סקרי מסמך דרישות <i>SRS</i>	תיכון וארכיטקטורה – הגדרות, דוגמאות נפוצות, <i>UML</i> , סדנת תיכון משימת פרויקט 4: כתיבת מפרט תיכון – <i>SDS</i> סקרי מסמך דרישות <i>SRS</i>	4
<i>Firestore</i> , התחברות למסד נתונים בענן ואוטנטיקציה	ניהול פרויקט תוכנה – הערכה, תכנון, שיטות וכלים, <i>Scrum</i> , <i>Github Issues</i> , יצירת <i>Backlog (project)</i> סקר <i>SDS</i> משימת פרויקט שלב 5: סבב 0 – <i>ZFR</i>	5
השלמות <i>back-end</i> השלמת סקרי <i>ZFR</i>	בקרת תצורה \ קוד – <i>Revision Control</i> כלים: <i>git / github</i> משימה אישית 3 סקר <i>ZFR</i> משימת פרויקט 6: תחילת סבבים – סבב 1 <i>MVP</i>	6
פגישות וסקרי סבב	בקרת תצורה ב - שיטות: מודלי ענפים	7
"	בקרת תצורה ג – בקשות משיכה השלמת משימה 3 סקרי סבב 1 - הצגה, סקר קוד, משימות רטרוספקטיבה ותכנון לסבב 2	8
"	בדיקות – מבוא, בדיקות יחידה, פיתוח מונחה בדיקות (<i>TDD</i>), כלי בדיקה משימה אישית 4: <i>TDD</i> ו-בדיקות יחידה	9
"	בדיקות 2 – פיתוח מונחה בדיקות	10
"	בדיקות 3 – טיפול בתלויות, <i>mock objects</i> , כיסוי קוד ונושאים מתקדמים הרצאה אורח	11
"	תחזוקה וקוד קיים (<i>legacy code</i>), מבוא לעקרונות <i>OOP</i> ותבניות	12
"	תחזוקה 2 משימת אישית 5	13
"	סיכום הקורס מצגות פרויקטים <u>אירוע סיום הנדסת תוכנה בשרות הקהילה</u>	14

- מרצה אורח מהתעשייה
- נושאים נוספים, לפי הזמן: תיכון מפורט, תיעוד, קוד פתוח, אבטחה ועוד

קורסים קשורים:

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

חובות וצינונים:

- הסטודנט נדרש להירשם למערכות הקורס המקוונות ולוודא שמעודכן בתדירות בפורום. מומלץ להביא מחשב נייד להרצאה – לפי הזמן נתרגל עבודה על הפרויקט והמשימות האישיות בהקשר לחומרי ההרצאה.
- פרויקט - 70%, עבודת צוות, הציון מורכב משלבים שונים שיימסרו במהלך הקורס במאגר חומרי הקורס, כגון: הצעת פרויקט, מפרט דרישות, מפרט תיכון וסבבי פיתוח וגם הצגות בהרצאה. בכל שלב יפורטו התאריכים לביצוע (הגשה בד"כ בחצות לפני ההרצאה הבאה) ומרכיבי הציון השונים כגון: רלוונטיות ואיכות המימוש, תיעוד הנדסי, משוברים וסקרים ועבודה עם לקוחות. הציון הוא קבוצתי אך יתכנו תיקונים משמעותיים לפי הערכת עמיתים (בסיום כל סבב לכל הצוות כולל המעריך לפי הודעה בנושא), תרומה לפרויקט, משוב מהלקוח והתרשמות צוות הקורס.
- משימות אישיות - 10%, 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~~. בסיום הסמסטר, אם ברצונכם לשמור את התרגיל נא להעתיקו מחשבון הקורס אליכם.
- מבחן – 18%, בעיקר על חומרי ההרצאה, חובת מעבר, תיתכן אפשרות תרומת שאלות למבחן רב-ברירה וכן פטור ע"י כתיבה ופרסום שטף של סיכום כיתתי ע"י נציגה.
- תרומה לקורס – עד 2%, השתתפות פעילה בהרצאה ובפורום (ללא קשר לפרויקט הצוותי) במהלך הסמסטר יפורסמו ציוני ביניים (לא סופיים) ומשובים בהקדם האפשרי.

איחורים: במקרה של סיבה מוצדקת בלבד יש לתאם מראש לו"ז חלופי באישור המרצה ומתועד במייל, תיתכן הורדת ניקוד בהתאם לנסיבות ולזמן - עקב עומס המשימות בקורס לא נוכל לבדוק עבודות שיוגשו באיחור.

יחידה: 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