

מודלים לפיתוח מערכות תוכנה Software Systems Modeling														
<table border="1"> <tr> <td>הרצאה</td><td>תרגיל</td><td>מעבדה</td><td>נקודות זכות (נ"ז)</td><td></td></tr> <tr> <td>3</td><td>0</td><td>0</td><td>3.0</td><td></td></tr> </table>					הרצאה	תרגיל	מעבדה	נקודות זכות (נ"ז)		3	0	0	3.0	
הרצאה	תרגיל	מעבדה	נקודות זכות (נ"ז)											
3	0	0	3.0											
12003				קוד הקורס										
○ הנדסת תוכנה (10014)				דרישות קדם										
אין				דרישות מקבילות										

מטרות הקורס

בקורס נלמד:

- מבוא למידול
 - טכניקות מידול (modeling) בסיסיות ופורמליות
 - הכרות עם כלים
 - פיתוח יכולת אבחנה ותיעדוף בין רמות שונות של הפשטה
 - פיתוח יכולת פתרון בעיות וניתוח אנליטי ע"י מידול סוגים שונים של מערכות תוכנה
 - פיתוח החשיבה לגבי מידול יעיל ואפקטיבי לצרכי תיכון (design), מימוש, אימות ותיקוף של מערכות תוכנה
- הקורס ילווה בתרגילים ופרויקט בהם ייבחנו ויישמו העקרונות השונים.

נושאי הקורס כוללים

UML, (Simulink, Alloy, Z, MDE), דוגמאות (case studies) למידול תוכנה
הצגת מחקרים בנושאים קרובים של פיתוח מונחה מודלים, בדיקות, הנדסה הפוכה (reverse engineering) ועקיבות.
ייתכנו (=צפויים) שינויים והתאמות במהלך הסמסטר.

חובות וציונים

אין חובת נוכחות למעט בהצגות נדרשות, אך יינתן בonus 5% על השתתפות פעילה.
ש.ב. - 60%
פרויקט סופי - 40%
אפשר להגיש בקבוצות של 2-3 סטודנטים. כל קבוצה נדרשת לפחות פעם אחת במהלך הסמסטר להציג את עבודתה בסקר כיתתי.
את כל המודלים יש ליצור באמצעות כלי מידול רלוונטיים (ולא בכתב יד) ולהגיש תוצרים בעזרת מערכת בקרת גרסאות.
פרויקט (תכנית כללית):
שלב 1 – פתרון של אחד מרשימה של אתגרי מידול
שלב 2 – מימוש המודל של קבוצה אחרת
ייבחר צוות מנצח ע"י המרצה ואולי שופטים נוספים.

מרצה

ד"ר ראובן יגל, robi@post.jce.ac.il, שעות קבלה – בתיאום, פורום, שם משתמש ב-gh+bb:robi-y

ספרות

אין (כרגע) ספר קורס נדרש, יימסרו חומרי קריאה וקישורים למדריכים במהלך הסמסטר.

ספרות מומלצת

- (1) M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3e, Addison-Wesley (2003).
- (2) M. Brambilla, J. Cabot, M. Wimmer, *Model-Driven Software Engineering in Practice*, Morgan & Claypool (2012).
- (3) R.S. Pressman, *Software Engineering: a Practitioner's Approach*, 7th ed., McGraw-Hill (2010).
- (4) I. Sommerville, *Software Engineering*, 9th ed., Addison Wesley (2011).
- (5) E. M. Clarke, Jr., O. Grumberg & D. Peled, *Model Checking*, MIT Press (1999).
- (6) D. Peled, P. Pelliccione, P. Spoletini, "Model Checking", In: Wiley Encyclopedia of Computer Science and Engineering, Wiley (2009).
- (7) B. Berard ... [et al.], *Systems and Software Verification: Model-Checking Techniques and Tools*, Springer (2001).