

## סילבוס קורס

### יסודות הנדסת תוכנה

### 4021374

פרטי הקורס

**שנה אקדמית:** תשפג  
**סוג הקורס:** חובה  
**רמת הקורס:** תואר ראשון  
**צורת העברה:** פנים אל פנים, Project oriented.  
**דרישות קדם:** מבוא למדעי המחשב 4080021  
**דרישות במקביל:**  
**שפת הוראה:** עברית  
**סביבת עבודה:**  
**מתרגלים:** מר אלון טפליצקי  
[AlonTa4@ac.sce.ac.il](mailto:AlonTa4@ac.sce.ac.il)  
מר אושרי אברהם  
[OshreAv@ac.sce.ac.il](mailto:OshreAv@ac.sce.ac.il)  
מר אריאל אפשטיין  
[ArielEp@ac.sce.ac.il](mailto:ArielEp@ac.sce.ac.il)

**קמפוס:** באר שבע  
**מחלקה:** הנדסת תוכנה  
**תחום:**  
**שנת לימוד:** ב'  
**סמסטר:** א  
**נקודות זכות:** 4  
**נקודות ECTS:** 6  
  
**מרצה/ים:** ד"ר הדס חסידים  
[hadasch@sce.ac.il](mailto:hadasch@sce.ac.il)

מטרה

לימוד יסודות הנדסת תוכנה כולל מושגים בסיסיים, ותהליכי פיתוח תוכנה, הכרת שיטות ומתודולוגיות ליישום הידע בפרויקט תוכנה, דגש על ייזום ועיצוב חווית משתמש, הנדסת דרישות, תקשורת ועבודת צוות במסגרת ביצוע הפרויקט המעשי. מתודולוגיית מפל המים ושיטות אגיליות.

## תפוקות למידה

- עם סיום מוצלח של הקורס, הסטודנטים יהיו מסוגלים:
1. להסביר את מושגי היסוד, מדדים והשלבים העיקריים בפיתוח תוכנה בדגש על הגישה האג'ילית.
  2. לנתח דרישות וליישם עקרונות עיצוב ומידול בסיסיים.
  3. ליישם שלבי הפיתוח של פרויקט תוכנה בסביבה האג'ילית.
  4. להסביר מהי ארכיטקטורת מערכת וסוגי מערכות שונים.
  5. לתכנן את שלבי הפיתוח להגעה לתוצר הרצוי כולל עיצוב FLOW (מסמכי ייזום, עיצוב, קוד, אימות ותיקוף).
  6. ליישם מיומנויות "רכות" בעבודת צוות ותקשורת בפרויקט תוכנה.
  7. לאמת ולתקף את הקוד באמצעות תהליכי איכות ובדיקות יחידה.

## תוכן הקורס

שבוע	נושא	מקורות רלוונטים
1	מהי הנדסת תוכנה, מבוא להנדסת תוכנה, פיתוח מוצרי תוכנה ודרישות הקורס [הרצאה]	[1] פרק 1 [2] פרק 1
2	SDLC, מסמך ייזום-product charter-vision, הנדסת דרישות מיצוי וכלים [הרצאה]	[1] 2 [2] 3
3	אג'יל, תקשורת ועבודת צוות [הרצאה], פגישות הנחיית פרויקטים	[2] 2 [3] 7 [4]
4	עיצוב והיבטים סוציו-טכניים של מערכות [הרצאה]	[1] 10.4, 10.2 [3] 1,3
5	פגישות הנחיית פרויקטים	
6	בוחן 1 פגישות הנחיית פרויקטים	
7	האקטון תוכנה עובדת	
8	פגישות הנחיית פרויקטים ארכיטקטורה-עצמאי	[1] 5,6 [2] 4
9	בדיקות ומדדי איכות [הרצאה]	[1] 24, 8 [5] 8.3, 3
10	פגישות הנחיית פרויקטים	
11	הערכה ואיטרציות [הרצאה]	[3] 9
12	בוחן 2, פגישות הנחיית פרויקטים	
13	הצגת פרויקטים [פרונטאלי]	

## מקורות ספרות נדרשים ומומלצים

- ספר הקורס:
1. [Sommerville \(2015\), Ian. Software Engineering. 10th Edition. Harlow: Pearson Education Limited](#)
  2. [Sommerville, I. \(2020\). Engineering software products. London: Pearson](#)
  3. [Knight, W., Knight, W., & Corrigan. \(2019\). UX for Developers. Apress](#)
- מקורות נוספים:
4. [Zuill, W., & Meadows, K. \(2016\). Mob Programming: A whole Team Approach. In Agile 2014 Conference, Orlando, Florida](#)

5. *Naik, K., & Tripathy, P. (2011). Software testing and quality assurance: theory and practice. John Wiley & Sons*

### פעילויות למידה מתוכננות ושיטות הוראה

שעות הרצאה שבועיות: 3, שעות תרגול שבועיות: 1, שעות מעבדה שבועיות: 1. מספר מפגשים פרונטליים: 4-7. הקורס נלמד בגישת Project Oriented. בשבוע הראשון לסמסטר יתקיים מפגש פתיחה בו יוצגו דרכי העבודה והדרישות כולל חלוקה לקבוצות (צוותים של 4 סטודנטים). כמחצית מההרצאות יתקיימו באופן פרונטלי. התרגול יתבצע במעבדת מחשבים להנחיה ותרגול. בנוסף, לקבוצות יוגדרו מפגשי הנחייה אישיים. במהלך הסמסטר יוגשו חלקים של הפרויקט לפי לוח מוגדר מראש ובתום הסמסטר יוגש הפרויקט. כמו כן במסגרת הקורס תתקיימו תחרות האקטון, בסופו נדרשים הסטודנטים להגיש תוצר קוד עובד.

### שיטות הערכה וקריטריונים

קריטריון	אחוז	הערות
בחנים:	20%	בוחן 1 הינו רשתי – הוא מהווה מגן לבוחן 2 כך שמי לא עובר את בוחן 1 ציונו ייקבע על פי בוחן 2 בלבד. מי שניגש ועבר את שני הבחנים, ציונו יהיה הגבוה מבניהם.
תרגילים:	20%	מסמך ייזום, דרישות ותכן (17%), בדיקות יחידה (3%).
פרויקט:	45%	הגשה פרויקט סופית (45%) כולל הצגה פרונטלית
האקטון:	15%	לקראת אמצע הסמסטר - האקטון יישומי.
נוכחות:		חובת נוכחות בכל ההרצאות והתרגולים. 2 נקודות יופחתו מן הציון הסופי עבור כל היעדרות.