



המחלקה למדעי המחשב סילבוס קורס אקדמי

שם הקורס: **עקרונות שפות תוכנה** מספר הקורס: **150060**
היקף הקורס: **שעור: 3 ש"ש תרגיל: 1 ש"ש מעבדה: 1 ש"ש נ"ז: 3.5**

תיאור קצר של הקורס:

מטרת הקורס היא להקנות ידע על עקרונות פרדיגמות תכנות (תכנות אימפראטיבי, פונקציונלי, מונחה עצמים, לוגי, ומקבילי), ולהדגים באמצעות שפות המיישמות אותן.

מטרת הקורס:

תכנית הינה טקסט המתארת תהליך חישובי מסויים (אלגוריתם). שפת תכנות הינה שפה מלאכותית שבמסגרתה אפשר לכתוב טקסט תכנותי. פרדיגמה מדעית הינה אוסף תיאוריות ושיטות מקובלות לקהילה המדעית בתקופה היסטורית מסוימת. באמצעות תיאוריות ושיטות אלו המדע מסביר ומנבא תופעות. פרדיגמה תכנותית הינה אוסף הנחות, תיאוריות, שיטות ותקנים, המגדיר מונחים כגון: מה זה חישוב?, איך מתבצע חישוב?, איך (כלומר, באמצעות איזה כלים) מייצגים מידע? וכו'. כל שפת תכנות קשורה לפרדיגמה תכנותית מסוימת, והקשר הזה קובע במידה רבה את התחביר ואת הסמנטיקה של השפה. עקרונות ייחודיים לפרדיגמת תכנות מסוימת, באים לידי ביטוי בשפות הקשורות אליה; למעשה, שפת תכנות מיישמת פרדיגמת תכנות מסוימת, לפעמים יותר מפרדיגמה אחת. מצד שני, יש עקרונות תכנות מסוימים שמופיעים בצורות שונות בשפות הקשורות לפרדיגמות שונות.

לכן, מטרת הקורס הזה לספק לתלמידים ידע על עקרונות פרדיגמות תכנות (תכנות אימפראטיבי, פונקציונלי, מונחה עצמים, לוגי ומקבילי), ע"י אנליזה והנגדה של מבני שפה: מה הם האפשרויות העומדות למעצבי שפה, כיצד ממומשים המבנים, וכיצד האופן הראוי לשימוש במבנה מסויים. ידע זה יאפשר לתלמידים (א) לזהות את פרדיגמת התכנות הקשורה לשפה מסוימת, וכך לסווג את השפה הזאת בהתאם לפרדיגמה שלה; (ב) לבחור פרדיגמה נכונה (ובהתאם, שפת תכנות) על מנת לפתור בעיה בצורה הפשוטה ביותר.

בפלטפורמות תכנות עכשוויות, דוגמת .NET ו Java, מקובל להשתמש בארכיטקטורה של שתי שכבות (CLR ו JVM, בהתאמה). לכן, כמטרה העומדת בפני עצמה, נלמד להכיר ארכיטקטורה זו, ונשתמש בה לצורך בניית קומפיילר המכיל בתוכו מימוש של חלק מהנושאים התיאורטיים של הקורס.

נושאי הקורס (החלק התאורטי - 3 ש"ש):

מבוא (מה הם שפות תכנות, קריטריונים להערכת שפות, היסטוריה של שפות, וכד'), אלמנטים של שפת תכנות, ופרדיגמות תכנות: אימפראטיבי (Imperative), מונחה עצמים (Object oriented), פונקציונאלי (Functional), לוגי (Logic), ומקבילי (Concurrent). מטרת ארכיטקטורת שתי השכבות, והשלבים לבניית קומפיילר על מנת להכיר את מושגי הסינטקס, סמנטיקה, ארכיטקטורת מחשבת, יחידות הניתוח הדקדוקי ויצירת קוד (הסמנטיקה של השפה).

חובות ומטלות:

1. במשך הסמסטר יחולקו 5 תרגילים תכנותיים מהספר של Nisan&Schocken (מקור 4).
א. התרגילים ניתנים להורדה מאתר הספר (<http://www.nand2tetris.org>). הגשתם תהיה בקצב של אחד לשבועיים.
ב. שפת התכנות: i.
בתחילת הסמסטר יחולקו שפות (ש[כנראה] אינכם מכירים) לכל זוג בקורס.





- ii. עליכם ללמוד שפה זו על מנת לפתור את כל התרגילים (מלבד השלישי). פיתרון התרגיל יוגש בשפה שחולקה לכם.
- ג. לא תתאפשר הגשה מאוחרת על תאריך ההגשה שיקבע.
- ד. התרגילים בנויים אחד על השני, כך שלבסוף יהיה לנו קומפילר לשפה דמוית java. בכל אופן, אי הגשה של תרגיל אינה מונעת הגשת התרגילים האחרים. ולכן, אף אם לא עמדתם במטלה מסוימת, אפשר להמשיך עם המטלות הבאות ועדיין להפיק תועלת.
- ה. אופן ההגשה והבדיקה של התרגילים יפורטו ע"י המתרגל(ת).
- ו. יש לעשות את התרגילים **בזוגות** (בלבד).
2. ניתוח והצגת שפת תכנות
 - א. כפי שנאמר דלעיל, בתחילת הסמסטר יחולקו שפות (ש[כנראה] אינכם מכירים) לכל זוג בקורס.
 - ב. עליכם ללמוד שפה זו, כך שתוכלו ..
 - i. לתת סקירה היסטורית לגבי הצורך בשפה (מה היא באה לפתור), הרקע והידע הטכנולוגי שהיה בתקופת פיתוחה, ומידת השגת מטרותיה
 - ii. לסווג אותה על פי הפרדיגמות שיוצגו בקורס במשך הסמסטר.
 - iii. לנתח כיצד היא מתמודדת עם בעיות עיצוביות שונות.
 - iv. להציג את ממצאיכם בהרצאה לפני הסגל בתאריך שיקבע (לקראת סוף הסמסטר).
 - עליכם:
 1. להכין מצגת שתשמש אתכם להרצאה זו.
 2. להצטמצם לסדר גודל של כחצי שעה (קחו בחשבון שזמן הרצאה ממוצע לשקף הוא בין דקה אחת לשתי דקות).
 3. לענות על שאלות שיוצגו לפניכם ע"י הסגל ו/או סטודנטים אחרים.
3. בחינה מסכמת

שים לב: חומר הבחינה יכלול ג"כ מאמרים, קישורים, בלוגים, וכו' שיפורסמו באתר הקורס אשר יש **חובה** לקוראם

מדדים להערכה:

תרגילים (30%), הצגת שפה (40%) ובחינה מסכמת (30%).

בונוס בבחינה (אופציונאלי):

בעת עבודתכם בשפה שחולקה לכם, מן הסתם תתקלו בבעיות (כמו בכל שפה חדשה כשמתחילים לעבוד איתה). **אם ברצונכם ב 10% בונוס בבחינה**, עליכם לתעד בויקי הנמצא באתר הקורס את הבעיות ופתרונן (כיצד התגברתם על התקלה – הפיתרון והמקור לתשובה). הבונוס ינתן על כתיבה משמעותית בויקי, ולא על שינויים מינוריים על מה שכבר כתוב. שינוי משמעותי יכול לכלול (ותלוי בשיקול דעת סגל הקורס):

- i. הוספת ערך לשפה שאינה נמצאת בויקי
- ii. איחוד, סידור ותיקון ערכים כפולים
- iii. וכו'
- iv. **כמובן, שערכים שאין בהם משהו משמעותי לתקן, לא יזכו את הסטודנטים בנקודות**

נוכחות: בתרגילים חובה (עם אפשרות הסנקציות הנמצאות בתקנון)

ציון הצגת השפה יחושב על סמך המרכיבים הבאים:

- (א) ציון על המצגת
- (ב) ציון על אופן הצגת הדברים בהרצאה





(ג) ציון על אופן ואיכות התשובות לשאלות הסגל במשך הרצאת הצגת השפה
 (ד) ציון על הדגמת אחד מחמשת התרגילים - במעמד הצגת השפה מול הסגל, תיכונת הדגמת
 אחד מחמשת התרגילים שנעשו במשך הסמסטר.
 ← על כל אחד הנ"ל ישאלו שאלות אשר ישפיעו על הציון.

קורסי קדם:

153105 תכנות פונקציונלית ולוגית

150015 מבנה נתונים ותכנות א'

150018 סדנא ב-C++

קורס מקביל: 151131 קומפילרים ומתרגמים

אתר הקורס: במערכת moodle

ביבליוגרפיה

1. Sebesta, R.W.: *Concepts of Programming Languages*, 6th Ed., Addison-Wesley, 2003.
2. Sethi, R.: *Programming Languages - Concepts and Constructs*, 2nd Ed., Addison-Wesley, 1996.
3. MacLennan, B.: *Principles of Programming Languages: Design, Evaluation, and Implementation*, 3rd Ed., Oxford University Press, 1999.
4. Nisan, N. and Schocken, Sh.: *The Elements of Computing Systems – Building a Modern Computer from First Principles*, MIT Press, 2005.
5. Henri Bal, Dick Grune: *"Programming Language Essentials"* ., Addison-Wesley, 1999.
6. Friedman, D., Wand, M. and Haynes, C.: *Essentials of Programming Languages*, MIT Press, 1992.
7. Ernest C. Ackermann, J. Ogden, *"Essentials of C Programming Language"* Research & Education Assn, 1998.
8. Sangal, R.: *Programming Paradigms in LISP*, McGraw Hill College, USA, 1991.
9. Harold Abelson, Gerald Jay Sussman, Julie Sussman: *Structure and Interpretation of Computer Programs (SICP)*, MIT Press, 2nd ed., 1996.
10. Sterling, L., Shapiro, E.: *The Art of Prolog, Second Edition: Advanced Programming Techniques (Logic Programming)*, MIT Press, 1994.
11. M. Ben Ari, E.: *Understanding Programming Languages*, John Wiley & Sons, 1996.

