

## הקדמה

### מטרת הקורס

הקורס "סדנה ב – C" נועד ללמד אותכם את שפת התכנות C ולאפשר לכם להתנסות בתכנות פרוצדורלי שאינו מונחה עצמים. במהלך הקורס תלמדו לעבוד עם מצביעים, להקצות זיכרון, לשחרר זיכרון, ולעבוד עם זיכרון מוקצה. באופן כללי, התכנות שלכם יהיה "קרוב יותר למכונה" במובנים הבאים:

- תתאפשר לכם גישה ישירה לזיכרון המחשב. בעזרת גישה זו תוכלו לבצע פעולות במהירות רבה אך גם לגרום לנזקים חמורים בלי לשים לב.
- בשעת התכנות ב – C תצטרכו להיות מודעים לתהליך הקומפילציה.
- מכיוון שאתם תתכנתו בסביבת Unix ומכיוון שמערכת הפעלה זו ו-C כורכות זו בזו, בתכניות שלכם תהיה לפעמים התייחסות מפורשת למערכת ההפעלה.

בעבר נלמד חומר זה (או דומה לו) בקורס שניקרא "programming lab" או PLab. מלבד לימוד השפה, אחת המטרות העיקריות של PLab הייתה התמודדות עם מטלות תוכנה לא טריוויאליות, הכוללות טיפול במקרי קצה שונים. למרות שהתמודדות עם מטלות תוכנה בעלות משקל היא מטרה חשובה, הוחלט השנה לשנות את תכנית הלימודים וללמד קורס המתמקד בפרדיגמת התכנות ואינו מתיימר לתרגל תכנות מסיבי.

מכיוון שתכנות הוא מקצוע הנדסי, מובן שאי אפשר להיחפז למתכנת מיומן ללא תרגול אינטנסיבי וללא התמודדות עם המורכבות של מטלות תוכנה גדולות. תרגול זה יתקיים בקורסי התכנות של המשך לימודיכם.

### מדוע חשוב ללמוד C גם כיום ?

C היא שפת תכנות פשוטה מאד שהומצאה בתחילת שנות השבעים. לכאורה ניתן לטעון ששפה בת למעלה משלושים שנה, שלא עברה שינוי משמעותי מאז המצאתה, היא שפה ארכאית שאין עניין ללמוד אותה כיום. מסתבר שגם כיום יש כמה סיבות טובות ללמוד ולשלוט ב – C. הסיבה העיקרית היא שהשפה עדיין נמצאת בשימוש נרחב, עדיין הרבה אנשים בוחרים לכתוב ב – C את התוכנה שלהם והרבה מהתוכנה הנמצאת בשימוש, כתובה ב – C. רב התוכנות בהם תשתמשו בקורס כתובה ב – C (לדוגמה: מערכת ההפעלה – GNU/Linux, העורך – emacs והקומפיילר –(gcc-ים)). כאשר רוצים לשנות או להוסיף מאפיינים לתוכנה הכתובה ב – C, צריך כמובן לדעת את השפה.

C שפה פשוטה ואלגנטית. קל לכתוב לה קומפיילר וקל לתרגם אותה לקוד שפת מכונה יעיל. כיום C היא כנראה השפה שבה ניתן לתכנת על הכי הרבה פלטפורמות וגם לקבל ביצועים הדומים לשפת מכונה. סיבה חשובה נוספת ללימוד השפה היא שהשפה מהווה בסיס לשפה פופולרית נוספת הנקראת ++C. ++C היא למעשה הרחבה של C ולכן כל תכנית ב – C היא גם תכנית ב ++C (אבל לא להפך).

אתם לומדים C לאחר שלמדתם שני קורסים ב – Java. זה כמובן לא הסדר ההיסטורי של הדברים. באופן גס אפשר להגיד ש – C הומצאה בתחילת שנות ה – 70, ++C, בתחילת שנות ה – 80 ו – Java בתחילת שנות ה – 90. כל שפה הושפעה מהשפות שקדמו לה. התחביר שאתם מכירים מ – Java דומה מאוד לתחביר של ++C והתחביר של ++C הוא הרחבה של התחביר של C. בגלל ההשפעות האלה, התחביר של C יראה לכם מוכר מאד. למרות זאת יש להיזהר, מכיוון שלעיתים הדמיון החיצוני מסתיר הבדלים עקרוניים בין השפות.

### כמה מילים על הנדסת תוכנה ולימוד תכנות

תחום הנדסת התוכנה הוא תחום צעיר, בערך בן חמישים או שישים שנה בלבד. אתם חיים בתקופה מאד מיוחדת בהיסטוריה, תקופה בה בפעם הראשונה יש בידי אנשים כלי כללי המאפשר לממש לוגיקה כלשהיא. בעבר, עבור כל טכנולוגיה חדשה שהומצאה, היה צריך להמציא מנגנון הממש את הלוגיקה שלה. לדוגמה, אם בעבר רצו שמנוע המכונית יקבל תערובת דלק התלויה במהירות שלו, היה צריך להמציא מנגנון מכני הממש זאת. היום, ניתן להשתמש

במעבד המסוגל להריץ סידרת כללית של הוראות ולכתוב עבורו סדרת הוראות מסוימת המתארת את הפתרון לבעיה. האפשרות החדשה הזאת, לפתור בעיות בעולם האמיתי ע"י כתיבה של קוד, טומנת בחובה הבטחות גדולות. חלק מההבטחות האלה אפשר כבר לראות מקוימות היום. בימינו, המחשב השתלט כבר על הרבה תחומים. לדוגמה:

- חישוב ופתרון בעיות.
- תקשורת.
- תכנון וייצור בעזרת מחשב.
- גרפיקה ועיבוד תמונה.
- עריכה וסידור הדפסה.
- מוזיקה (יצירה, עריכה והשמעה).
- סרטים (יצירת אפקטים, הצגה).
- משחקים.
- מכירה ותיווך.

קיימים עד תחומים רבים נוספים. מבחינה טכנולוגית אנו נמצאים בעיצומה של מהפכה שנראה שהאנושות עד לא יודעת בדיוק איך לעכל. בעתיד אולי יקראו לתקופתנו "מהפכת האינפורמציה" או "מהפכת התקשורת" או "המהפכה החישובית" (גם בתקופת המהפכה התעשייתית לא קראו לה "המהפכה התעשייתית"). כטבעם של מהפכות, המאופיינות בתנודות חריפות ושינויים קיצוניים, כך גם הטכנולוגיה היום עוברת תהפוכות רבות. נראה שתחום הנדסת התכנה עדיין רחוק מלהגיע לבשלות. יש תחושה של בוסריות במערכת המושגים ובמונחים של התחום. עדיין לא ברור לאף אחד איך צריך ללמד תכנות. יש טענה שלימוד תכנות צריך להתבצע בצורה של שוליה ומומחה, אחרים טוענים שפשוט צריך להתמודד לבד עם מטלות תוכנה רבות ויש האומרים שצריך לקרוא הרבה קוד שנכתב ע"י אחרים. כנראה שרצוי שילוב בין כל השיטות האמורות בתוספת קורסים והשתלמויות.

**"הוא עבד ברצינות, זאת חצי אמנות"**

**וחצי עבודת נמלים**

**הוא נתן את כולו, גם הצוות שלו**

**לא נתן מנוחה לכלים"**

(מתוך "התעשייה האבירית" – מאיר גולדברג וקורין אלאל)

בשביל אלה מכם שבחרו ללמוד לתואר ראשון במדעי המחשב זהו קורס חשוב במיוחד. רצוי שבוגר תואר במדעי המחשב ירגיש נוח לקבל מטלת תוכנה מוגדרת ולתרגם אותה לקוד איכותי. קוד איכותי הוא קוד יעיל, יציב, בהיר, קל לתחזוקה וניתן להרחבה ללא צורך בשינוי רב. התרגילים בקורס הם ההזדמנות שלכם לתרגל ולשייף את הטכניקה שלכם. למרות שהתרגילים יהיו קצרים וממוקדים, עדיין ניתן להפיק מהם הרבה. אחת לכמה זמן תהיה לכם אפשרות לביצוע מטלת תכנות יותר מאתגרת במסגרת תרגיל רישות. למי שיש זמן, מאד מומלץ לנסות להתמודד עם מטלות אלו. כדאי להתייחס לקורס כהזדמנות להשתפר כמתכנתים ופחות להתייחס לציון בו. ההשפעה של הציון על החיים שלכם בהמשך תהיה כנראה קטנה בהרבה מההשפעה שתהיה להשתפרות שלכם כמתכנתים. תכנות יכל היות דבר מאד מהנה. כאשר אתם מגיעים לדרגת מיומנות מסוימת, התחושה היא של יכולת וחופש יצירה גדול.

**"תכנות זה כסף"**

(אמרה עממית)

**על אתר הקורס**

אתר הקורס אמור לשמש אותכם באופן אינטנסיבי במשך כל הקורס. האתר יכיל את כל החומר של ההרצאות בצורת סיכומים מילוליים. אין למעשה כל צורך לסכם את ההרצאה בכיתה. מכיוון שהחומר מופיע בצורה מסודרת באתר, ההרצאות ירשו לעצמם לעתים להיות קצת אסוציאטיביות ולגלוש לנושאים צדדיים. כמו כן קיים באתר מעין לוח מודעות ציבורי בו תפורסמה הודעות שוטפות הנוגעות לקורס (ההודעה החדשה ביותר, בתחילת הרשימה). כדאי לבדוק את לוח המודעות באופן תדיר. כל מסמכי האדמיניסטרציה, נמצאים באתר במרוכז. תרגילי התכנות והפתרונות שלהם יפורסמו באתר במהלך הקורס (התרגיל הראשון כבר נימצא). לאנשים המעוניינים בקריאה נוספת, קיים דף המכיל המלצות שונות.