



משרד החינוך
מינהל למדע וטכנולוגיה



המרכז הישראלי להוראת המדעים
ע"ש עמוס דה שליט



האגף לתכנון ופיתוח
תכניות לימודים, משרד החינוך



מרכז המורים הארצי למדעי המחשב



מכון ויצמן למדע



הטכניון - מכון טכנולוגי לישראל

סקר ספרות

כלים ושיטות להוראת מדעי המחשב

צוות פיתוח:

שרה פולק וד"ר צביקה פירסט, מט"ח

יצא לאור במימון האגף לתכנון ולפיתוח תכניות לימודים במשרד החינוך

ומטה המרכז להוראת המדעים ע"ש עמוס דה שליט

© כל הזכויות שמורות למשרד החינוך

מורה יקר,

לפניך סקר ספרות שנועד לחשוף בפניך כלים ושיטות הוראה במדעי המחשב, היבטים שונים של המחקר והשלכותיו על הוראת מדעי המחשב.

סקר הספרות יתמקד בסקירה של תיאוריות ושיטות הוראה, כלים וסימולציות המתאימים למדעי המחשב. כל אחד מהכלים והשיטות מיועד לשיפור הוראת המקצוע בהיבט פדגוגי ספציפי.

סקר הספרות יוקדש לשלוש הגישות הבאות: למידה בגישה הקונסטרוקטיביסטית, למידה מבוססת בעיות, ומינימליזם.

בכל גישה תוכל למצוא התייחסות לתיאור הגישה או השיטה, יתרונות וחסרונות פדגוגיים, הצעות פרקטיות למימוש הגישה בתכנית הלימודים במדעי המחשב. כמו כן תוכל למצא בסוף החוברת קישורים למקורות ואתרים נוספים העוסקים בגישות אלה.

תוכן עניינים:

3	מבוא
4	פרק 1. על למידה בגישה הקונסטרוקטיביסטית
4	1.1 תהליך למידה
4	1.2 פדגוגיה בגישה הקונסטרוקטיביסטית
6	1.3 הגישה הקונסטרוקטיביסטית ומדעי המחשב
8	פרק 2. למידה מבוססת בעיות - problem-based learning
8	2.1 מבוא- מהי למידה מבוססת בעיות
9	2.2 מודל ההוראה
11	2.3 מחקרים במדעי המחשב על למידה מבוססת בעיות
14	2.4 דוגמה ליישום גישה ללמידה מבוססת בעיות בקורס יסודות 1
19	פרק 3. מינימליזם
19	3.1 מבוא – מהו מינימליזם
21	3.2 יישום מינימליזם במדעי המחשב
21	3.3 מינימליזם ותכנות מונחה עצמים
23	3.4 קרל הרובוט כסביבה ליישום מינימליזם בתכנות מונחה עצמים
24	3.5 יישום הגישה המינימליסטית ביחידת הלימוד "מבוא למערכות מידע"
28	פרק 4. סיכום תמציתי של המאמרים
29	4.1 לקראת הוראה קונסטרוקטיביסטית: ברוקס וברוקס
30	4.2 קונסטרוקטיביזם בהוראת מדעי המחשב: בן ארי
32	4.3 מינימליזם: קרול
34	4.4 תכנון מונחה עצמים: רובסון וקרול
35	4.5 למידה מבוססת בעיות: סבורי ודאפי
37	4.5 למידה מבוססת בעיות במדעי המחשב: קיי וחבריו
40	רשימת מקורות

מבוא

הוראה הוא מקצוע דינאמי העוסק בפיתוח חומרי למידה ובתכנון ויישום מערך הוראה מתאים בכיתה. אולם למרות התכנון הקפדני, המפגש היום-יומי בין מורים ותלמידים, יוצר מצבים הדורשים מהמורה גמישות ויכולת אילתור. לא פעם עומדים המורים בכיתה מול קושי או שאלה שלא נתנו עליה את הדעת בתכנון השעור, או רעיון גישה חדשה להצגת נושא שצף כתגובה למשפט של תלמיד. בסיום השיעור וניתוח של מהלך השעור, קיימים לא פעם ספקות ותהיות האם באמת הושגו מטרות השיעור. תופעות ומצבים אלו הם שכיחים ומורים חווים אותם יום יום במקומות ותחומי דעת שונים. חלק מהתופעות קשורות לתהליכי למידה והוראה וניתן להסבירן באמצעות תיאוריות שעוסקות בנושאים אלו. נגדיר תיאוריה כמערך מאורגן של רעיונות ושיטות המספק מסגרת לפרשנות של עובדות וממצאים. לדעתנו הכרות עם תיאוריות של למידה והוראה, תסייע לנו בבואנו להתאים את ההוראה למקצוע אותו אנו מלמדים ובהתמודדות עם מצבים שונים בכיתה. בסקר זה נציג תיאוריות הוראה שניתן ליישמן ביחידות הלימוד השונות במדעי המחשב.

הפרק הראשון מתייחס ללמידה והוראה על פי הגישה הקונסטרוקטיביסטית, המתארת למידה כתהליך בו הלומד בונה באופן אקטיבי את הידע שלו על סמך התנסויותיו וידע קודם. כמו כן נציג עקרונות של הפדגוגיה הנגזרת מעקרונות הגישה הקונסטרוקטיביסטית. בסיום הפרק נציג שני מאמרים המציגים את ההיבט הקוגניטיבי וההיבט החברתי. המאמרים מתארים את הקשיים ביישום הגישה הקונסטרוקטיביסטית במדעי המחשב ובהתאם מתארים המלצות מתאימות להתאמת ההוראה ב-CS.

בשני הפרקים הבאים נציג שתי חלופות "לתכנון" של הוראה. במילים אחרות נציג שתי גישות להוראה: למידה מבוססת בעיות ומינימליזם. בחרנו בשתי גישות אלו משום שלדעתנו הן מספקות מסגרת מתאימה לפיתוח חומרי למידה והוראה המדגישים את מטרות הוראת מדעי המחשב הבאות: תלמיד ירכוש את היכולת לפתרון בעיות באופן עצמאי יחד עם הבנה של המושגים והעקרונות של מדעי המחשב. בתיאור הגישות השונות, נציג את המאפיינים של הגישה ועקרונות של מודל ההוראה הנגזרים ממנה. לאחר מכן נתאר מחקרים ודוגמאות ליישום הגישה בהוראת יחידות לימוד שונות במדעי המחשב.

בפרק השני נציג את הגישה **ללמידה מבוססת בעיות**, המתוארת כשיטת לימוד המציבה את הלומד בתפקיד אקטיבי של פותר בעיה מחיי היום-יום, וכך מסייעת ללומד לפתח כישורים של פותר בעיות עצמאי יחד עם רכישת ידע בתחום דעת. בניגוד לגישה המסורתית, בה ארגון הקורס הוא סביב נושאים ותכנים בתחום הדעת והמורה במרכז תהליך למידה ומשמש כמקור ידע, בלמידה מבוססת בעיות התלמידים במרכז תהליך הלמידה, בו הם מתמודדים עם פתרון בעיה ובתהליך זה הם רוכשים את הידע הדרוש. נתאר יישום של הגישה להוראת תכנות מונחה עצמים ולהוראת קורס מבוא במדעי המחשב (יסודות 1).

בפרק השלישי נציג את **המינימליזם**, כתיאוריה שהתפתחה בשנות 1980 והיא מתארת את השלכות של שימוש בסביבות מחשב על למידה ומתארת מודל להכנת חומרי למידה לעבודה עצמית (תיעוד תוכנות ופיתוח מדריכים למשתמש). למינימליזם יש הרבה משותף עם הקונסטרוקטיביזם. שתי התיאוריות הללו מתייחסות ללמידה אקטיבית המאפשרת ללומד להבנות את הידע של עצמו תוך כדי התנסויות ועל בסיס ידע קודם. בחלק השני של הפרק נתאר יישום של הגישה לתכנון ועיצוב מערכות מידע והכנת חומרי למידה לקורס בתכנות מונחה עצמים וליחידת הלימוד "מבוא למערכות מידע".

פרק 1. על למידה בגישה הקונסטרוקטיביסטית

1.1 תהליך למידה

למידה מוגדרת כשינוי התנהגותי הנובע מהתנסות שחווה הלומד במגע עם העולם (תופעות, אנשים, אובייקטים). השינוי ההתנהגותי יכול להיות גלוי (שינוי בהרגלי האכילה למשל) או שינוי סמוי (בקוגניציה כמו ידע, בעמדות במחשבות וברגשות). קיימות תיאוריות רבות המתארות מודל לתהליך הלמידה אולם אנו נתרכז בתיאוריה הרווחת כיום והיא הקונסטרוקטיביזם.

על פי התיאוריה הקונסטרוקטיביסטית, הלמידה היא תהליך בו הלומד בונה באופן אקטיבי את הידע שלו לגבי העולם שבו הוא חי. הידע שצבר הלומד מאורגן בסכמות מורכבות וקשורות זו בזו. למידה משמעותה, חיפוש כלים שיעזרו לנו להבין את התנסויותינו. כאשר אנו נתקלים ברעיון, אובייקט או תופעה חדשה, אנו מנסים לפרש את ההתנסות החדשה ולהתאימה לידע או מערך הכללים שצברנו עד כה. הידע הקיים הוא הבסיס שעל פיו אנו מפרשים את ההתנסות החדשה, והפרשנות שאנו צוברים מוסיפה ומשנה את הידע הקיים. ככל שאנו לומדים יותר הפרשנות שנתנו לתופעות דומות בעבר משתנה ומתעדכנת בהתאם להתנסויות החדשות (ברוקס וברוקס, 1997). במילים אחרות, הידע של הלומד נוצר בתהליך של הטמעה בה הוא מעצב ידע חדש למבני ידע קיימים ובתהליך של התאמה, הלומד משנה את הסכימה כך שתתאים לידע החדש שצבר. בן-ארי (2001) תיאר תהליך זה כרקורסיבי, בו ידע קיים מסייע לבנות ידע חדש שהופך להיות הידע הקיים.

גורם נוסף שמשפיע על תהליך הלמידה הוא ההיבט החברתי. חוקרים טוענים כי ידע נבנה בין אנשים וילד בתחילת חייו לומד מהוריו ומהסביבה הקרובה לו. כך שניתן לומר שתהליך למידה כרוך גם בהפנמה, כלומר בתהליך של בניה של ייצוג קוגניטיבי פנימי של פעולות פיזיות או מנטליות שתחילתן באינטראקציות חברתיות.

נסכם את המאפיינים של למידה לפי הגישה הקונסטרוקטיביסטית:

- למידה היא תהליך אקטיבי ומתמשך.
- למידה מתבצעת תוך כדי בנייה של ידע ע"י הלומד.
- הלמידה היא רקורסיבית כשידע והתנסויות חדשים מפורשים על סמך ידע והתנסויות קודמים ובסוף תהליך של התאמה והטמעה הוא הופך לידע הקיים.
- ידע הוא סובייקטיבי הנוצר מפרשנות של הפרט להתנסויות שלו עם העולם.
- למידה היא תהליך קוגניטיבי וחברתי.

1.2 פדגוגיה בגישה הקונסטרוקטיביסטית

ישום תיאוריה זו בבתי ספר מציבה בפני הפדגוגים אתגרים לא מעטים (ברוקס וברוקס, 1997):

- לפי הגישה הקונסטרוקטיביסטית, למידה היא תהליך בניה של הבנות על העולם האמיתי. אולם, בית ספר הוא מתווך בין לומד העולם האמיתי ללומד ומהיותו כזה, הוא אינו יכול לספק את כל העושר והמורכבות של העולם. יתרה מזאת, הלמידה מתבצעת באמצעות התיווך של מורים וחומרי למידה. ההוראה כמתווכת יוצרת עולם מושגים ותופעות שהן ייצוג של העולם אותו מנסים להעביר לתלמידים, והתלמידים מפרשים ייצוג זה ויוצרים לעצמם את הייצוג שלהם.
- בעיה נוספת נובעת מכך שבית ספר בוחר את הנושאים והמידע שתלמיד ילמד, ובכך מרחיק את הלומד מהקונטקסט. כלומר הידע נלמד בסביבה סינתטית ולא בסביבה האמיתית.
- בית ספר מנסה ליצור אחידות הן בנושאים הנלמדים והן בצורה הנלמדת וזאת בניגוד לגישה הקונסטרוקטיביסטית המדגישה את המגוונות והסובייקטיביות של הידע אצל הלומד.

- קושי נוסף נובע מכך שלמידה והוראה מאורגנים בתכנית לימודים שהיא סטטית במהותה. תכנית הלימודים נבנת לכל מקצוע והיא כוללת את הנושאים והמושגים ואפילו את סדר הצגתם בכיתה. גישה זו אינה מייצגת נאמנה את התופעות בעולם האמיתי משום שהיא לא מאפשרת הצגה הוליסטית של תופעות שהן מטבען מורכבות וכוללות תחומים שונים.
- בבית ספר, המורה במרכז והוא בעל הידע הקובע מה חשוב ומה לא, והמוביל את תהליך הלמידה. סביבת למידה כזו הופכת את הלומד לפחות אקטיבי ויותר פסיבי בלמידתו.

ז'קלין ברוקס ומרטין ברוקס (1997) טוענים שהתאמת פדגוגיה לגישה הקונסטרוקטיביסטית מושתתת על חמישה עקרונות:

1. העלאת בעיות בעלות רלוונטיות ראשונית לתלמידים ותפקיד המורה הוא לא העברת ידע אלא תיווך וסיוע לתלמיד בבניית ההבנות של הנושאים הנלמדים.
 2. הבניית הלמידה סביב "רעיונות גדולים" או מושגי יסוד. תוכנית הלימודים מוצגת "מן השלם אל החלק", תוך הדגשת המושגים הרחבים והקשר בין תחומי. ההנחה היא שטבען של השאלות המוצגות לתלמידים משפיע במידה רבה על עומק החיפוש של התלמידים אחר תשובות. שאלה רחבה מאפשרת לתלמידים נקודות כניסה מרובות. היא מזמינה כל תלמיד להשתתף בפתרון, בדרכו הייחודית. מכאן מובן שכל נושא נלמד הוא רב תחומי כי רק בכזה מתקיימת חקירה אמיתית.
 3. ביקוש וייחוס ערך לנקודות מבט של תלמידים. דרישה זו קשורה ישירות לפילוסופיה של הקונסטרוקטיביזם הטוענת שאין אמת מוחלטת וכל דבר הוא יחסי. על כל דבר אפשר להסתכל מנקודות ראות שונות ולהבין אותו באופן שונה. המורה הקונסטרוקטיביסטי צריך להכיר ולהוקיר את נקודות הראות השונות של תלמידיו ולהעריכן.
 4. התאמת תוכניות לימודים כך שתתייחסנה לידע המוקדם של תלמידים. במילים אחרות, תוכנית הלימודים והמורה צריכים להתחשב ביכולות שהתלמיד מביא עמו לכתה - יכולות קוגניטיביות, חברתיות ורגשיות.
 5. הערכת הלמידה צריכה להיות בהקשר משמעותי לתלמיד ולהתייחס לבעיות שהתלמידים נתקלים בהן תוך כדי למידה. אסור לה להיות שיפוטית. מבחנים לסוגיהם אינם מתאימים לגישה הקונסטרוקטיביסטית. למעשה ההערכה בגישה זו היא חלק מההוראה והלמידה ממשיכה גם בזמן ההערכה.
- Fosnot (1996) מוסיפה את ההפשטה הרפלקטיבית כאחד העקרונות של הפדגוגיה הקונסטרוקטיבית. לדעתה זהו הכוח המניע את הלמידה, כיוון שבני אדם שואפים לארגן ולהכליל את התנסויותיהם בצורה של ייצוג מופשט כלשהו. מתן זמן לחשיבה, ייצוג בצורה רב-סמלית ודיון בקשרים בין התנסויות, עשויים לדעתה לאפשר הפשטה רפלקטיבית.

לסיכום, בסביבה קונסטרוקטיביסטית הכיתה היא קהילת לומדים המעורבת בפעילות, תוך כדי שיח וחשיבה משותפים על נושאים משמעותיים ללומדים. המורים מעודדים ומקבלים אוטונומיה של תלמידים ויוצרים אווירה נוחה להתבטאות. תשומת לב רבה ניתנת ללומד: לידע הקודם שלו, לניסיונו, לאסטרטגיות החשיבה שלו ולמוטיבציה, ובהתאם לכל אלה מנווט המורה את הנחייתו של התלמיד. סביבת למידה קונסטרוקטיביסטית חשוב שתהיה עשירה ככל האפשר ורב תחומית, ושתספק לתלמידים הזדמנויות רבות ככל האפשר לחיפוש משמעות, להערכה של אי-ודאות ולנטילת אחריות על למידתם. חשיבתם של התלמידים היא המניעה את השיעורים ולכן דיאלוג, חקר והתלבטות זוכים להערכה.

1.3 הגישה הקונסטרוקטיביסטית ומדעי המחשב

יישום הגישה הקונסטרוקטיביסטית במדעי המחשב נחקרה רבות. בסעיף זה נציג שני מאמרים המציגים את ההיבט הקוגניטיבי וההיבט החברתי. המאמרים מתארים את הקשיים ביישום הגישה הקונסטרוקטיביסטית במדעי המחשב ובהתאם מתארים המלצות מתאימות להתאמת ההוראה ב-CS.

בן-ארי (2001) מתייחס ללמידה כתהליך קוגניטיבי של הפרט הבונה את הבנתו על סמך התנסויות ופרשנות של העולם על בסיס ידע קודם. במאמרו טוען בן ארי שבהוראת מדעי המחשב יש לקחת בחשבון שני גורמים חשובים המשפיעים על למידה:

1. לתלמיד מתחיל אין מודל מנטלי אפקטיבי של מחשב שהוא חיוני לפרשנות ובניית ידע חדש על בסיס התנסויותיו בכיתה.
2. המחשב יוצר מציאות אונותולוגית נגישה וקל לקבל תשובה שיכולה להתפרש כ"נכונה" ולכן כדי להשיג ביצועים מוצלחים יש לבנות של מודל נורמטיבי של המציאות.

בן ארי מצרף מספר המלצות לתכנון הוראה ב-CS :

1. יש ללמד מודל של מחשב המתייחס למבנה חומרה וזרימת מידע בו בצורה מפורשת. השאלה היא מה רמת פירוט הדרושה ולכך יש מספר תפישות החל מפירוט הרכיבים האלקטרוניים וכלה בויזואליזציה של מושגים כמו העברת פרמטרים. ההיקף קשור כמובן למטרות ההוראה, והוא יבחר כך שהוא יתמוך בהבנה של תהליך ביצוע תכנית על המחשב.
2. מודל חיוני של מחשב צריך להילמד לפני האבסטרקציה, גם בשפות מונחות עצמים (הרמה בה אנו מתכנתים בשפות אלו היא מופשטת ורחוקה מאוד מן המחשב שברמה התחתונה).
3. אסור שהמציאות המפתה של המחשב תתפוס את המקום של בניית המודל המנטלי.
4. צריך להכיר בעובדה שיש גישות שונות לתכנות שנוגעות מתכנון מלמעלה למטה ועד בריקולז. בריקולז היא גישה לתכנות (שתוארה על ידי Turkel and Papert) הנתנת לתיאור מלמטה למעלה, או תכנות על ידי "מימוש" והתנסות. כלומר קודם יש התנסות עם העצמים ואז יש בניה של מושגים ועצמים חדשים. בן-ארי טוען שכולנו משתמשים ברמה כלשהי של בריקולז, אבל שימוש בבריקולז ללא מודל מנטלי מתאים והבנה טובה של המושגים והפרקטיקה יכולה להפריע לתלמידים מתחילים.
5. משימות מעבדה מאפשרות לתלמיד להיות אקטיבי ולבנות ידע על בסיס התנסות ולכן דרושה תשומת לב רבה לתכנון נכון של פעילויות מעבדה. לטענת בן-ארי, עדיף לשלב משימות סגורות וסוג המשימות צריך לעודד תהליכים קוגניטיביים כמו ניתוח, בחינת אפשרויות וקבלת החלטות.
6. תשומת לב צריכה להנתן לשיטת הערכה. בניגוד למבחנים בהם מעריכים תוצאות, הערכה במדעי המחשב צריכה להיות לא רק לתוצר אלא בעיקר לתהליך של פתרון הבעיות.

בן-דוד קוליקנט (2004) יוצאת מהגישה הסוציו-תרבות, טוענת שלא רק קוגניציה משפיעה על למידתו של הלומד, אלא גם תרבותו ומכאן שאותו ידע קוגניטיבי יכול להיות מופעל אחרת בתרבויות שונות. תפישה זו נשענת על משנתה של ג'יין לייב, כי למידה היא חברתית. למידה, על פי לייב ווגנר, היא תהליך של כניסה לתרבות (enculturation), בו הלומדים הם חדשים המנסים להתקבל לקהילה של מומחים. הלומדים עוסקים בפעילויות פריליות, ותוך כדי התנסות הם רוכשים ידע וניסיון ההופכים אותם לחברים מלאים בקהילה. לטענת בן-דוד קוליקנט, תלמידים במדעי המחשב הם אורייני מחשב שפיתחו שיטות עבודה ומושגים בסיסיים כמשתמשי מחשב, והם תופשים את עצמם

"כמומחים במחשבים". לעומת זאת, הוראת מדעי המחשב בבית ספר הארץ מציגה את התרבות של האקדמיה המדגישה לא שימוש במחשב אלא חשיבות של תיאוריות ומושגים. המפגש בין שתי התרבויות, של התלמידים אורייני המחשב ושל המורים נציגי האקדמיה, בכיתת מדעי המחשב יוצרת "התנגשות" בין תרבויות. התלמידים אינם נותנים לגטימציה לנלמד בכיתה ולפעולות הלימודית בהם הם נדרשים לעסוק. כתוצאה מכך הם אינם "מטמיעים" את הידע והפרקטיקה אותה אנו מקנים להם, ומכאן לא לומדים את רזי המקצוע. כפתרון להתנגשות בין תרבויות, מציעה בן-דוד קוליקנט את מודל הוראה שנקרא "אזור למידה פורה", והוא כולל פעילויות שהן רלוונטיות לתלמיד וכך תקבלנה את הלגיטימציה שלו, אך הן כוללות גם את הידע של קהילת היעד. כך נעודד את התלמיד לעסוק בפעילות, ללמוד ולהבין את החשיבות של הידע של תרבות היעד. כך, בהוראה המבוססת על התנסות בביצוע אוסף של פעילויות שבהן גדל המינון והמורכבות של תרבות קהילת היעד, התלמיד חוצה את הגבול מתרבותו כמשתמש מחשב אל התרבות של קהילת אנשי CS ורוכש את הידע הפרקטיקה הרצויים. בן-דוד קוליקנט מנסחת מספר המלצות ברוח זו לגבי הוראת מדעי המחשב:

- הכרת התרבות של התלמידים ועיצוב משימות שתהיינה משמעותיות עבורם. משימות שותקבלנה את הלגיטימציה של התלמידים, תעודד אותם לעסוק בהן. לעומת זאת, משימות שהתלמידים לא מוצאים אותן שימושיות לתרבות שלהם, לא זוכות בלגיטימציה ומביאות להתנגשות. לדוגמה בעיצוב תכנה הבדל תרבויות עשוי לגרום להתנגשות. כמורים, אנו מציגים את ההיבט התיאורתי בו עיקר העיסוק בטיפוסי נתונים מופשטים (ADT), הוא תכנון ADT מתאים וזיהוי ההזדמנות להשתמש בו. תלמידים לעומת זאת מפרשים זאת כמיותר ולא מובן, (או כפי שתיאר אחד התלמידים: "כשהגעתי לעיצוב תכנה שמחתי. חשבתי יופי עכשו נעצב מסכים ונכתוב תכניות יפות. במקום זה עסקתי בתכנון ממשק לרשימות ועצים ולא כל כך יודע מה לעשות אתם"). דוגמה אחרת היא השימוש בבעיות מפורסמות במדעי המחשב, בעיות להן יש סיפור כיסוי דימוני כמו בעיית הסועדים הפילוסופים. תלמידים המחשיבים קונקרטיים לא מבינים למה שימוש בבעיות כאלה טוב, ולכן יש להתחיל במשימות קונקרטיים ובהמשך להכלילן לסיפור הכללי.
- מורים צריכים להיות מודעים לפירושים ופרקטיקה של התלמידים שצברו כמשתמשי מחשב השונים מן הפרקטיקה והמושגים של תרבות האקדמיה ב-CS. לדוגמה, מושג נכונות בעיני תלמידים היא נכונות יחסית, ולכן תכנית עובדת (ולא בהכרח תמיד) היא תכנית נכונה. בעוד במדעי המחשב יש הגדרה ברורה ומוחלטת של תכנית נכונה, בה לכל קלט מתקבל הפלט הנכון.
- הערכה של התלמידים לתכניות עשירות בפירוטכניקה (אם אפשר גם אם ומים) והערכה של התלמיד לטכנולוגיות ושפות תכנות חדישות המאפשרות יצירת תכניות כאלה. לכן רצוי לעצב סביבות למידה שמאפשרות גם פירוטכניקה וגם מדגישות את הצגת המושגים והעקרונות של מדעי המחשב.

פרק 2. למידה מבוססת בעיות - problem-based learning

2.1 מבוא – מהי למידה מבוססת בעיות

Finkle and Torp (1995) הגדירו למידה מבוססת בעיות (בקיזור PBL) כשיטת לימוד המציבה את הלומד בתפקיד אקטיבי של פותר בעיה מחיי היום-יום, וכך מסייעת ללומד לפתח כישורים של פותר בעיות עצמאי יחד עם רכישת ידע בתחומי דעת שונים. גישה זו פותחה לפני כ-30 שנה ע"י Barrows (1986) בבית ספר לרפואה (באוניברסיטת McMaster בקנדה), בעקבות קשיים של סטודנטים לרפואה לרכוש את הידע והניסיון הדרוש לאבחון מקרים קליניים. בניגוד לסביבת למידה מסורתית שהייתה נהוגה, לפיה ארגון הקורס הוא סביב נושאים ותכנים בנושא מסוים והמורה במרכז תהליך למידה ומשמש כמקור ידע, Barrows תכנן קורס בו הסטודנטים התמודדו עם ניתוח של מקרים קליניים אמיתיים ותוך כדי כך הם למדו את המושגים והעקרונות הרפואיים ורכשו יכולת פתרון בעיות. עקרונות גישה זו נגזרים מהתיאוריה הקונסטרוטיבסטית השמה דגש על בניית ידע אקטיבית וביקורתית על בסיס ידע קודם. בגישה זו התלמיד הוא במרכז ותהליך למידה מאורגן סביב בעיות אמיתיות, ותוך כדי התמודדות עם הבעיות, התלמיד מגדיר, אוסף ורוכש את הידע בתחומי הדעת (אליהם קשורה הבעיה), ובנוסף הוא רוכש את המיומנויות והניסיון בפתרון בעיות. במשך הזמן, גישה זו אומצה על ידי בתי ספר ואוניברסיטאות בתחומים אחרים.

כאמור, PBL היא גישה פדגוגית המאתגרת תלמידים "ללמוד איך ללמוד", תוך כדי שהם מחפשים פתרון לבעיות שהוצבו בפניהם (Savery and Duffy, 1995). התמודדות עם בעיות מעודדת חשיבה ביקורתית ואנליטית, ומספקת הזדמנויות לביצוע פעולות כמו איתור ושימוש במקורות ידע, ניתוח ותכנון פתרונות והערכתם. בנוסף, גישה זו מאפשרת התנסות ורכישת מיומנויות של עבודה שיתופית בקבוצה. לסיכום, למידה-מבוססת-בעיות מסייעת לתלמיד להשיג את היעדים הבאים:

- יכולת למידה של נושאים חדשים
- רכישת מיומנויות פתרון בעיות
- חשיבה יצירתית וביקורתית
- פיתוח גישה הוליסטית לבעיות ומצבים
- יכולת עבודה עצמית
- יכולת לעבוד בקבוצות תוך כדי שיתוף פעולה ושיפור יכולת תקשורת

ניתן לסכם בטבלה את ההבדלים העיקריים בין סביבת למידה בגישה המסורתית לבין סביבת למידה בגישת PBL

מסורתית	PBL
המורה במרכז תהליך הלמידה	התלמיד במרכז תהליך הלמידה
דגש על רכישת הידע	דגש על שימוש בידע לפתרון בעיות
ידע מאורגן בחלקים שלמידת כולם מהווה את השלם	הידע מוצג בשלמותו בהתחלה ואחר כך מפרקים אותו לחלקים
תפקיד מורה – מעביר ידע	תפקיד מורה – מסייע מנחה
ארגון למידה מובנה וקשיח	ארגון למידה גמיש לשינויים
מורה אחראי על למידת התלמיד	התלמיד אחראי על למידתו

טבלה מס. 1 הבדל בין סביבת למידה בגישה מסורתית ו בגישת PBL

2.2 מודל ההוראה

מודל של למידה-מבוססת-בעיות מתייחס לאופן ארגון הקורס, תכנון תהליך ההוראה, בחירת בעיות שסביבן תאורגן הלמידה, הכנת חומרי למידה והוראה, הגדרת תפקיד המורה והתלמיד, יישום והערכה של הישגי התלמידים ושל הקורס עצמו.

לקורסים המפותחים בגישה זו יש מספר מאפיינים בולטים:

- הבעיות שאיתן מתמודדים התלמידים הן בעיות מחיי היום-יום שמאפשרות התמודדות עם תהליך פתרון אותנטי ככל האפשר.
- בין-תחומיות: הנושאים והמושגים שבהם נוגעים בפתרון הבעיה, הם בדרך כלל קשורים למספר תחומי דעת.
- עבודה בקבוצות.
- תפקיד המורה הוא להנחות את הקבוצות ולסייע לתלמיד ולקבוצה בתהליך פתרון הבעיה.
- שימוש בבעיות שהן ill-structured פתרון בעיה דורש בדרך כלל איסוף מידע ולימוד נושאים ומושגים חדשים. התלמיד צריך לזהות את הידע הדרוש לו ובהתאם להשתמש במקורות ידע נוספים כדי לרכוש ידע זה.

פתוח קורס בגישה זו מבוסס על בחירת בעיות שסביבן יעוגנו כל הנושאים וכל הפעילויות הלימודיות. קיימות מספר גישות לפתוח קורס בגישה זו. החל מהגישה בה לאורך כל הקורס, הלמידה של תלמידים היא למידה עצמית ושיתופית תוך כדי התמודדות עם הבעיה שהוצגה בפניהם ושימוש בחומרי עזר ו- tutorials. במסגרת זו, תפקיד המורה הוא להנחות ולסייע בעבודה של התלמידים. מסגרת אחרת משלבת התמודדות עם בעיות והרצאות. במסגרת זו יש שילוב של עבודה עצמית של תלמידים המתמודדים עם הבעיה יחד עם הרצאות וחומרי עזר (המספקים ידע בתחום תוכן, דוגמאות של פתרון בעיות, tutorial). במקרה כזה תפקיד המורה הוא לא רק להנחות אלא גם לספק ידע.

תכנון בעיה מתאימה הוא קריטי והיא צריכה לכלול את המאפיינים הבאים:

- עיסוק בעיה מהעולם האמיתי.
- בעיה פתוחה שפתרונה אינו חד משמעי אלא מאפשר להתייחס למגוון של אפשרויות.
- דורשת למידה עצמית.
- דורשת עבודת צוות.
- התמודדות עם הבעיה מאפשרת להשיג יעדים לימודיים ולפתח כישורים של פתרון בעיות.
- הבעיה מבוססת על ידע והתנסות קודמים אך פתרונה דורש למידה עצמית של חומר חדש.

פיתוח קורס בגישה זו מכיל את השלבים הבאים:

א. זיהוי תוצרי הלמידה בהתייחס למטרות הלמידה: תכנים וידע בתחומי הדעת הנלמדים, כישורים ומיומנויות לפתרון בעיה (כמו איסוף מידע, ניתוח, סינתזה וכדומה), כישורים ומיומנויות חשיבה גבוהות, כישורים חברתיים (כמו כישורי תקשורת, תפקוד בקבוצה וכדומה).

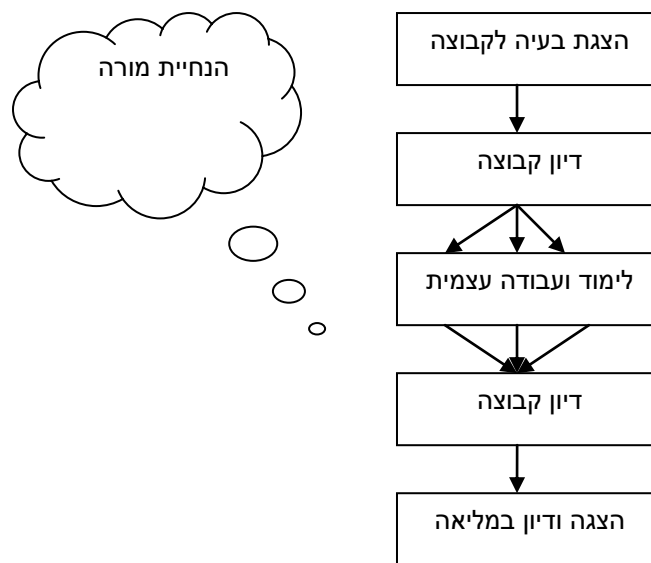
ב. הגדרת שיטת ההערכה של התלמיד מתייחס להגדרת התוצרים שיבדקו (בשלושת התחומים: ידע, מיומנויות פתרון בעיות ומיומנויות חשיבה גבוהות), הגדרת כלי ההערכה והקריטריונים והציונים לכל קריטריון. בנספח מצורפת דוגמה לכלי הערכה.

ג. תכנון הבעיות כולל את הפעולות הבאות:

- הגדרת המטרות הלימודיות הספציפיות שיושגו באמצעות הבעיה (ידע בתחום הדעת, מיומנויות פתרון בעיות ומיומנויות חשיבה מסדר גבוה).
- ניסוח הבעיה, כולל פירוט הקשיים צפויים.

- בדיקת הבעיה - האם הבעיה מאפשרת את השגת המטרות הלימודיות הספציפיות והתאמתה לפי הצורך.
 - הוספת מקורות מידע אפשריים.
 - הכנת חומרי עזר מתאימים.
- ד. תכנון מערך ההוראה (מערך השיעורים המוקצה לפתרון הבעיה). במערך שיעורים זה יש לתכנן את המפגשים כך שיאפשרו לתלמידים העוסקים בפתרון הבעיה, הזדמנות לעבוד ביחידים ובקבוצות. כמו כן, יש להקצות זמן לדיונים עם המורה במליאה (כיתה), זמן להערכה של תוצרי הביניים והתוצר הסופי.
- ה. היישום מתחיל מהצגת הבעיה בפני התלמידים וקביעת את "כללי המשחק". בשיעור זה יש לתאר את:
- המטרה והתוצרים הדרושים
 - שיטת עבודה וחלוקה לקבוצות
 - שיטות הערכה
 - לו"ז
 - הגדרת תפקיד התלמיד ותפקיד המורה.

ניתן לתאר את תהליך העבודה של התלמידים בעזרת האיור הבא. תהליך העבודה של התלמידים מלווה בהנחיה של מורה הניתנת בכל שלב ושלב לפי הצורך ובהקניית ידע דרוש.



איור מס. 1 תהליך עבודה של תלמיד וקבוצה

ו. לסיום, לאחר שהתלמידים סיימו לעבוד על הבעיה, יש לבצע הערכה של הבעיה ולבדוק האם היא השיגה את המטרות שנקבעו לה ואם לא כיצד ניתן לשפרה.

2.3 מחקרים במדעי המחשב על למידה מבוססת בעיות

מחקרים רבים בהוראת מדעי המחשב מראים כי לשימוש בגישה זו יש יתרונות רבים ולכן מספר סיבות: מדעי המחשב במהותו כרוך בפתרון בעיות אלגוריתמיות וגישה זו מאפשרת התנסות בתהליך פתרון. סיבה נוספת נובעת מהדינאמיות של המקצוע המדגישה את יכולת הלימוד עצמי של נושאים חדשים כמיומנות בסיסית. כמו כן עבודה במקצוע דורשת יכולת לעבודה בקבוצות, וגישה זו מספקת את ההזדמנות לכך (Ellis, A. et. Al., 1998). בסעיף זה נתאר מחקר שערכה קבוצת חוקרים מאוניברסיטת סידיני באוסטרליה. החוקרים השתמשו בגישת הוראה זו להוראת קורס מבוא בתכנות מונחה עצמים (Kay, J., et al., 2000). במאמר המסכם מחקר של מספר שנים, הם מציגים את הסיבות שהניעו אותם לבחור בגישה זו, מתארים את מבנה הקורס שפתחו ומציגים ממצאים ממחקר המציג את ההישגים של תלמידיהם שלמדו בגישה זו. תחילה החוקרים מפרטים את הקשיים העיקריים שהיו בהוראת המקצוע מבוא למדעי המחשב בגישה הקונבנציונלית:

- סטודנטים לא רכשו את המיומנויות לפתרון בעיות ולכן התקשו בהתמודדות עם בעיות. למרות שבתחום מדעי המחשב שרוב עיסוקו בפתרון בעיות, הוכח כי הידע הטוב ביותר נרכש תוך כדי התנסות והתמודדות עם בעיות.
- סטודנטים לא אמצו את הפרקטיקה של איש מקצוע במדעי המחשב. לדוגמה, קשה היה לשכנע תלמידים לתעד, בעיקר משום שתכניות בקורס מבוא היו קצרות.
- היו קשיים פדגוגיים שנבעו מכך שהידע של הסטודנטים שהגיעו לקורס היה טרוגני וסטודנטים בעלי רקע וידע קודם שונה במדעי המחשב שנאלצו ללמוד ביחד.
- בגישה הקונבנציונלית לא הייתה התייחסות לעבודה בקבוצות.
- קושי אחר נבע ממערך ההוראה הקונבנציונלי בו יש חלוקה לשיעורים פרונטליים ושיעורים מעשיים, כתוצאה מכך לא היו לתלמידים מספיק הזדמנויות לעסוק בפתרון בעיות אמיתיות, אלא רק בבעיות קטנות ומלאכותיות שניתן לפתור אותן בשיעור אחד.

החוקרים החליטו לבצע שני שינויים מהותיים בקורס המבוא: א. לשנות את הפרדיגמה התכנותית ולעבור משפה פרוצדוראלית (שפת פסקל) לתכנות מונחה עצמים (OOP). לשנות גם את גישת ההוראה ללמידה מבוססת פרויקטים. לטענת החוקרים, שבדקו גישות הוראה שונות, גישת ה-PBL מתאימה ביותר להוראת מבוא למדעי המחשב, משום שהיא מספקת הזדמנות לתלמידים לעסוק בפתרון בעיות אמיתיות ולרכוש תוך כדי התמודדות עם הבעיות, מיומנויות של פתרון בעיות וידע ב-OOP. בהמשך נתאר את ההמלצות והדוגמאות המתוארות במאמר ליישום הגישה בקורס OOP, אך תחילה נתאר בקצרה את תוצאות המחקר שערכו כדי להעריך את מידת התאמתה לתלמידיהם. לאחר שלוש שנים של מחקר ושיפור הקורס וחומרי למידה, מדווחים החוקרים כי ממוצע הציונים של התלמידים עלה ל-91 לעומת 63 (שלוש שנים לפני כן) בשאלות תכנות בסיסיות. בנוסף, לטענתם תלמידים שיפרו את היכולת שלהם בהבנת בעיות מורכבות יותר, כמו סיבוכיות, שקודם לכן נחשבו נושאים קשים מאוד ללמידה. שיפור נוסף חל בגישתם של התלמידים, במידת העניין בקורס ובהרגשה כי הם שיפרו את ידיעותיהם במהלך הקורס.

המלצות לתכנון קורס מבוא לתכנות ב-OOP

החוקרים מיפו את סוגי הידע שסטודנט המסיים קורס מבוא צריך לרכוש ובהתאם הם הגדירו את מטרות הלימודיות של הקורס, ובניהן המטרות הבאות:

- א. מטרות הקשורות בידע, בסיום הקורס הסטודנט יוכל:
 - להגדיר מישק של מחלקה פשוטה וליישמו.
 - להשתמש ביעילות בספרית מחלקות נתונה.
 - להשתמש באסטרטגיות קידוד ובדיקה הכוללת סגנון מתאים לכתובה של תנאים מקדימים ותנאי סיום (pre and post), הרצה ובחינה של מחלקה.

- להסביר ולהצדיק תכנון במונחים של שיטתיות, כלכליות ואת בחירת אסטרטגיה המתאימה לבדיקה של המחלקה.
- לקרוא ולהעריך של קוד מקור במונחים של מודולאריות, קוד עצמאי, מישק מחלקה וקשרים בין מחלקות.
- ב. שימוש בכישורים מטה קוגניטיביים:
- ידע בתכנון הכולל יכולת ניסוח פורמלי של הבעיות שיש לפתור, הגדרה של נושאים חדשים אותם יש ללמוד, הגדרה של אסטרטגיות ללמידת ידע חדש ויכולת בקרה והערכה של הידע שנרכש.
- הערכה עצמית הכוללת פיתוח אסטרטגיות המאפשרות בחינה של מושגי תכנות חדשים ושימוש בכלים שונים להערכה עצמית.
- הרפלקציה על תהליך הפיתוח, וקצב התקדמות ותוצרי הביניים.
- שימוש במדריכים ובחומר מודפס כדי למצוא מידע דרוש על תוכנות.
- יכולת לכתוב דוח על תכנון מחלקה ובחינתה המתאר את המטרות של כל בדיקה והסיבה לבחירה בה.
- יכולת להציג בעל פה בצורה מסודרת דיווח על הפתוח והיישום של המערכת שבנה.
- יכולת לעבוד בשיתוף פעולה, שימוש בכלים כמו תכנות-ע"י-חוזה (programming by contract) ויכולת תקשורת עם שאר חברי הקבוצה כדי להבטיח שכל אחד יודע את תפקידו ותחום אחריותו, נכונות לתרום למאמץ הכללי של הקבוצה ויכולת להעריך את תרומת הזולת.

ב. מבנה הקורס ב-OOP:

הסמסטר חולק לשלושה חלקים:

בעיה ראשונה (4 שבועות) - בחלק זה (startup) התלמידים מתוודעים לגישה של למידה מבוססת באמצעות פרויקטים והאחריות שמוטלת עליהם. לשם כך מוצגת בפני התלמידים בעיה איתה אותה הם צריכים לפתור. בארבעת השבועות ההרצאות בכיתה מתייחסות לתהליך הפתרון בעיה ותיאור של כלים מתאימים, הקניית ידע בסיסי במדעי מחשב הדרוש לפתרונה, והתייחסות מפורשת למיומנויות של עבודה בקבוצה. התלמידים מתנסים בביצוע מטלות שונות בהתאם לחומרים שפותחו על יד צוות ההוראה. הפעילות מסתיימת ברפלקציה על הבעיה שהתלמידים פתרו. למרות שהתלמידים עובדים בקבוצה, ההערכה בשלב זה היא אינדיבידואלית ומטרתה לתת משוב לכל תלמיד על טיב עבודתו.

בעיה שנייה (6 שבועות) בחלק זה (החלק המרכזי של הקורס) התלמידים בחרו בעיה אחת מתוך מגוון בעיות שניתנו להם. כדי לשפר את העבודה בקבוצה, המרצים עודדו יצירת קבוצות שבהן יש רקע שונה לחברי הקבוצה. המורים סיפקו שלבי ביצוע ולו"ז לביצוע של כל שלב. כמו כן בנוסף לעבודה הקבוצתית, נתנו הרצאות משלימות בתכנות מונחה עצמים ובהנדסת תכנה, וכל תלמיד נדרש להגיש חלק מהעבודה הקבוצתית וביחד עם עבודה מסכמת של הקבוצה. הערכה בשלב זה כוללת הערכה של ביצועי הפרט וביצועי הקבוצה. בנוסף לבעיה המרכזית, כל תלמיד צריך לבצע גם תרגילים נוספים (קצרים יותר) הקשורים לנושאים שונים.

סיכום והגשת דוחות והדגמות של פתרון בעיה 2 (שבועיים). בחלק זה של הקורס התלמידים מכינים דוחות ומצגות ומדגימים את הבעיות והפתרונות שלהם. חלק זה כולל גם הערכת עמיתים. הדוח שהם מגישים כולל לא רק את התוצר הסופי אלא את שלבי התכנון והעבודה.

ג. מקורות למידה

כדי לתמוך בתהליך הלמידה פותחו מגוון רב של חומרים ללמידה של נושאים ב-OOP ולפתרון בעיות. בנוסף החוקרים פתחו מאגר דוגמאות המתאר תהליך פתרון של בעיות מאותו סוג. הצגת הדוגמאות מאפשרת לתלמידים ללמוד מדוגמאות ולראות דרכים ואסטרטגיות מתאימות להתמודדות עם בעיות לדוגמה בחלק הראשון הבעיה הייתה לבנות סימולציה המסייעת למנהל בנק, ולכן נתנו לתלמידים דוגמה לבניית סימולציה של מערכת אקולוגית ודוגמה אחרת לבניית סימולציה של מעליות. חומרי עזר נוספים אותם פתחו החוקרים, הם דפים להערכה עצמית ודוגמאות לשימוש בהם.

ד. הכשרת צוות ההוראה

דגש רב הושם על הכשרה של צוות ההוראה כולל סמינרים ופגישות שבועיות, חומרי קריאה וכדומה.

ה. הערכה

תכנון שיטת הערכה כלל:

- קריטריונים המעריכים את הידע בתכנות מונחה עצמים.
- קריטריונים להערכה של תהליך הפתרון ושל התוצר הסופי.
- קריטריונים המעריכים את הכישורים בפתרון בעיות.
- קביעת משקל לפעולות השונות, ובהן מתן משקל שווה לצינונים של מבחנים ועבודה מעשית, ומשקל שווה לצינון של הפרט וצינון של הקבוצה.

דוגמה לבעיה שתוארה בקורס OOP:

מנהל בית התכנה הציג בפני הצוות אליו אתה משתייך פרויקט חדש – בניית סימולציה שתסייע למנהל בנק לבחון שתי שיטות לחישוב עמלות בגין פעולות של לקוחות בחשבון עובר ושב. גביית עמלות יכולה להיות מחושבת בדרכים שונות. שיטה אחת היא לגבות עמלה אחידה על כל פעולה שמבצע לקוח. אפשרות אחרת היא לגבות עמלות רק לסוג מסוים של פעולות מסוימות ומפעולות מסוג אחר לא לגבות כלל עמלה. ניתן לקבוע מדיניות לפיה יש מספר סוגים של לקוחות ולכל סוג לקוח להגדיר גובה עמלה. כדי לאפשר למנהלי הבנק לקבל החלטה לגבי השיטה היעילה לגביית עמלות, יש לבנות סימולציה שתציג ניתוח סטטיסטי של שתי שיטות שונות לחישוב עמלות והשפעתם על חשבונות של הלקוחות.

מנהל הבנק הגדיר מספר משימות שהסימולציה תכלול:

- עדכון פרטי לקוח הכולל מדיניות של גביית העמלה.
- הצגת פירוט הפעולות שבוצעו והעמלות שנגבו לכל פעולה.
- סיכום העמלות לפי סוגים שונים של פעולות וסוגים של לקוחות
- להוסיף מחלקה של טלר הכוללת מישק שיאפשר לבצע את הפעולות הבאות: לקדם כל משתמש בברכה, לשאול לסוג הפעולה המבוקשת: הפקדה/משיכה, לבצע את הפעולה ולדווח ללקוח בסיום על אחוז העמלה
- להוסיף מחלקת לקוח המכילה את פרטי הלקוח הדרושים לבנק.
- לשדרג את המחלקה של הטלר כך שבפעולת משיכה, יפורטו סוגי השטרות שנתנו ללקוח.
- להבטיח שלבנק יוכלו להיות הרבה לקוחות.
- להבטיח שללקוח יכולים הרבה חשבונות.
- להבטיח שללקוח יכול להעביר כסף בין החשבונות השונים שלו.
- להוציא דוחות חודשיים ללקוח.
- להוציא דוחות מסכמים להנהלת הבנק.

הנחיות ביצוע: המורה מיצג את תפקיד המנהל שלך. יש לעבוד בקבוצות של 3-4, כאשר כל תלמיד בקבוצה אחראי על חלק אחד של הסימולציה. משך הביצוע: 6 שבועות. כדי לעזור לך להתחיל, המנהל כבר שילם למתכנת בכיר שכתב מספר מחלקות (class) של בנק בהן תוכל להשתמש. במחלקות הן: חשבון לקוח, פעולה. הבעיה שניתנה לתלמידים כללה מפרט מדויק של מה שתלמיד צריך להגיש, ל"ז ותיאור של שיטת הערכה.

2.4 דוגמה ליישום גישה ללמידה מבוססת בעיות בקורס יסודות 1 במדעי המחשב

בסעיף זה נתאר בעיה המתאימה ללימוד נושאים ראשונים ביסודות 1 ומאפשרת להשיג את מטרות תכנית הלימודים (כפי שהוגדרה על ידי ועדת מקצוע) כ- "שילוב בין התאורטי-מושגי לבין המעשי-התנסותי באמצעות גישה דו-מסלולית שמלמדת יסודות מושגיים ומלווה לימוד זה בהמחשה (למשל מימוש האלגוריתם כתכנית בשפת תכנות והרצתה במחשב)". כדי להשיג מטרות אלו, בחרנו לבנות יישום - לומדה בחשבון לתלמידי גן. שימוש ביישום מסוג זה יאפשר לנו להציג תהליך הפתרון של בעיה אמיתית כפי שמבצעים אנשי מקצוע במדעי המחשב. לדעתנו שימוש ביישום מסוג זה מאפשר ליצר "אזור הוראה פורה" שאותו התלמיד ימצא מועיל משום שהוא עוסק בבניית מוצר ומצד שני מאפשר למורה להדגים עקרונות פתוח נכונים. השימוש ביישום, כולל תיאור היבט של המשתמש בלומדה (הילד והגננת) ותיאור היבט של מתכנת (התלמיד המפתח את הלומדה), וכך מאפשר לתלמיד להפנים את חובותיו והאחריות שלו לתכנית נכונה. באופן כללי אנו מגדירים ארבעה ממדי ידע (המתוארים בטבלה מס. 2), אתם תלמיד הלומד בגישה זו יצטרך לרכוש.

טבלה מס. 2: סוגי ידע בפתרון בעיות במדעי המחשב

<p>ידע שקשור לתחום הבעיה</p> <p>הגדרה של הידע הדרוש כדי לפתור את הבעיה, איסוף וניתוח מידע. שימוש במקורות וחומרי עזר הכוללים נתונים, מערכות דומות, tutorial</p>	<p>ידע במדעי המחשב</p> <p>פרוט הנושאים, המושגים והפעולות הקשור לתוכן או יחידות הלימוד</p>
<p>כישורים כלליים לפתרון בעיות (מתוך חלופות בהערכת הישגים / מנוחה בירנבוים)</p> <ul style="list-style-type: none"> • חשיבה לוגית המצאה/ יצירתיות, הבעה בכתב ובעל פה, הגדרת נושא, שאלת שאלות, שימוש יעיל במידע, עיבוד המידע, מיזוגו ואירגונו, ניסוח השערות למחקר, התבוננות/ תצפית, שימוש נכון במכשירים, ניתוח נתונים, הצגת נתונים מקושרים, הסקת מסקנות, שיפוט/ הערכה... • חשיבה ביקורתית, תכנון מהלך הפעולה, בחירת אסטרטגיה מתאימה, הערכת יעילות האסטרטגיה, הפעלת אסטרטגית תיקון, הערכה עצמית... • כשרויות חברתיות, ניהול משאבים ואיכויות אישיות: תקשורת עם הסביבה שיחה ודיון, שכנוע עבודת צוות, שיתוף פעולה, כיבוד הזולת. ניהול לוח זמנים, ניהול הצוות, ניהול המשאבים. מוטיבציה, סקרנות עמידה במצבי תסכול, נטילת אחריות, עצמאות, גמישות במצבים משתנים. 	<p>תהליך פתרון בעיה אלגוריתמית (הנדסת תכנה על קצה המזלג):</p> <p>שלבי פתרון בעיה ותכנון תכנית, ניתוח והגדרת דרישות, תכנון פתרון הכולל הגדרת: תת משימות, אלגוריתמים, מבני נתונים, קלטים/פלטים (תכנון מסכי קלט/פלט, מסננות קלט), מימוש בשפת תכנות, כתיבת מסמכי תיעוד, בחינה ובדיקה וקביעת קריטריונים להערכה של התכנה (יציבה, קלה לתחזוקה, ...)</p>

את ארבעת הממדים נתאים לבעיה אותה בחרנו להציג לתלמידים, אך תחילה נתאר את הבעיה.

תיאור הבעיה – פיתוח לומדה בחשבון לילדי גן:

בגן הילדים "עופרים" לומדים הילדים ביצוע פעולות חשבון בסיסיות (חיבור, חיסור, כפל וחילוק) על מספרים טבעיים מ-1 עד 100. בכל שנה מכינה רוחמה הגננת אוסף של דפים לתרגול אותם היא מחלקת לילדי הגן. לאחר שהם פותרים את התרגילים, היא בודקת אותם, וכך היא עוקבת אחר ההישגים של כל ילד. לאחר שנים רבות של עבודה בצורה זו, החליטה הגננת לשלב מחשב לתרגול של פעולות אלה. לכן פנתה לחברת תכנה וביקשה לומדה שתאפשר מגוון של הפעלות בחשבון. ההפעלות צריכות להיות מגוונות ולאפשר תרגול חוזר ונשנה עם נתונים חדשים. הלומדה תאפשר לילד לבחור את סוג הפעולה (חיבור/חיסור/...) ואת סוג ההפעלה, ובהתאם תספק לילד אוסף של תרגילים. הילד יוכל להכניס את תשובתו לכל תרגיל, התשובה תיבדק והילד יקבל משוב בהתאם. בנוסף כשילד מסיים את סט התרגילים בהפעלה, הלומדה ק משוב מסכם על כל התרגילים שביצע.

מנהל בית התכנה פנה לצוות אליו את משרתיך וביקש להכין לומדה מתאימה. על הצוות שלכם לפתח לומדה הכוללת 4 הפעלות מסוגים שונים. כל חבר בצוות חייב להכין הפעלה אחת. כל ההפעלות תאוחדנה למערכת אחת שתספק ממשק קלט/פלט להפעלה של הלומדה כולה.

בטבלה הבאה נתאר את ממדי הידע אליהם נתייחס בבניית הלומדה.

טבלה מס. 3: סוגי ידע הדרושים לתכנון לומדה לילדי גן

ידע במדעי המחשב (יסודות 1) משתנים, טיפוסים משתנים, הוראות קלט/פלט, הוראות השמה, חישובים אריתמטיים ובוליאניים, פונקציות ספריה, הוראות בקרה, שגרות פונקציות ופרוצדורות	ידע שקשור לתחום הבעיה • רקע בסיסי בחשבון • המושג לומדה: מה זו לומדה, מה היא כוללת, סוגים של לומדות וכדומה. • מקורות המכילים לומדות אותן ניתן לחקור: http://mail.edu-egev.gov.il/nitzan/nitzan_marpad/index.asp חיפוש במאגר בנושא מתמטיקה לגן וכיתה א, יניב אוסף רעיונות ללומדות.
הנדסת תכנה – ראו טבלה מס. 2	כישורים כלליים לפתרון בעיות – ראו טבלה מס. 2

הנחיות ושלבי ביצוע לתלמיד:

הנחיות כלליות:

1. עבודה היא בקבוצות של 4 תלמידים ומשך הביצוע 10 שבועות.

2. תהליך הביצוע:

א. ניתוח ואפיון ראשוני:

- בחנו סוגי הלומדות ופעילויות לתלמידי גן בהתאם למקורות המידע הנתונים
- אפיינו את צרכי המשתמש ותארו את סוגי ההפעלות שהלומדה תאפשר לתרגל
- הגדירו את דרישות הלומדה שלכם ואת תפקיד חברי הצוות
- הכינו מסמך מפרט דרישות (אפיון ראשוני)

ב. תכנון ועיצוב לומדה

- תכננו תכנון ראשוני של המודולים העיקריים, תפקידים ופעולות
- תכננו תכנון מפורט (תת משימות, מבני נתונים, אלגוריתם) של כל מודול.

ג. מימוש:

- כתבו את הקוד המתאים לתכנון של כל מודול הקפידו על תיעוד מתאים.
- בדקו שכל מודול עובד בצורה נכונה
- אחדו את כל המודולים ובדקו שכל הלומדה עובדת בצורה נכונה

ד. הכנת דוח מסכם:

- הכינו תיק תיעוד הכולל את אפיון המערכת, תכנון ראשוני ותכנון מפורט, קוד מקור והוראות הפעלה.
 - הכינו הצגה של הלומדה הכוללת מצגת על הלומדה.
3. בדקו לומדה של קבוצה אחרת ומלאו דף הערכה מתאים.

מערך השיעורים למורה:

הטבלה הבאה מסכמת את שיטת ארגון מערך השיעורים סביב הבעיה. (מספור אינו מציין שיעור ביחידות זמן אלא מגדיר את סדר הפעילויות). כמובן שכל מורה יכול להוסיף או לגרוע פעולות ולהתאימה לכיתה מסוימת.

טבלה מס. 4 מערך הוראה ליסודות 1 המאורגן סביב המשימה - פיתוח לומדה לילדי גן

מ.0.	הפעילות בכיתה	תכנים קשורים לפעילות	פעילות של קבוצות התלמידים
1.	<ul style="list-style-type: none"> • הצגת הבעיה בפני התלמידים ומתן הנחיות כלליות. • דיון ראשון בנושא לומדה, סוגי לומדות שהתלמידים מכירים, פעולות עיקריות 	<ul style="list-style-type: none"> • מה זה מדעי המחשב, מהו יישום • ישויות הקשורות ללומדה: משתמש, מתכנת, מה התפקיד של המתכנת, דרישות של לומדה. • בעיה אלגוריתמית והפתרון שלה כתכנית, הצגת שלבי פיתוח יישום 	איסוף מידע על לומדות ואפיון הלומדות הכולל מהם הפעולות העיקריות שהיא מבצעת.
2.	<ul style="list-style-type: none"> • הגדרת דרישות לומדה, הפעולות שתבצע, דרישות כלליות לנכונות, ידידותיות, תיעוד למשתמש. • תיאור תחום האחריות של התלמידים כמתכנתים • תיאור דרישות כלליות המתוארות בבעיה שהוצגה 	מאפיינים של בעיות אלגוריתמיות מרכיבים של תכנית: תת משימות, מבני נתונים ואלגוריתם	הגדרת דרישות הלומדה שהקבוצה תפתח. מוצר ביניים: מסמך מפרט אפיון הלומדה
3.	מרכיבי הלומדה - המודולים פעולות שכל מודול מבצע קשר בין מודולים השונים קשר בין מודולים והמשתמש - הגדרת קלטים/פלטים	תכנון ראשוני ופרוק משימה לתת משימה	הגדרת המודולים בלומדה תוצר ביניים: תיאור המודולים
4.	תכנון מסך פתיחה ומסך בחירה, קלט והצגת שם משתמש	הוראות קלט פלט, משתנים, עיצוב מסך, ידידותיות	עיצוב ומימוש מסך פתיחה עיצוב ומימוש מסך בחירה
5.	תכנון מפורט של מודול התרגול: תיאור תת-משימות	תת משימה, טענות כניסה ויציאה, פרוצדורות, מודולריות של תכנית - יתרונות	תכנון מודול התרגול בלומדה כל תלמיד בקבוצה יבצע פעולות 6 עד 10 על סוג ההפעלה שבחר.
6.	בניית תרגיל אחד	משתנים, הוראות השמה, ביטויים אריתמטיים, פונקציה של מספרים אקראיים, אלגוריתם מילולי	
7.	הצגת תרגיל, קליטת תשובה ומתן משוב	משתנים, הוראות קלט/פלט, הוראות השמה, ביטויים אריתמטיים, פונקציה של מספרים אקראיים, הוראות בקרה מותנת, אלגוריתם מילולי	
8.	הצגת סט של תרגילים	לולאת FOR, שימוש בצוברים של מספר תשובות נכונות, אלגוריתם מילולי	

9	הצגת משוב מסכם לסט תרגילים	צוברים, ביטויים אריתמטיים, הוראות קלט/פלט, אלגוריתם מילולי
10	שימוש במסוננות קלט	לולאה מותנית, ידירותיות, אלגוריתם מילולי
11	ניסוי ובחינה - testing	נכונות, שיטות לבדיקה
	אינטגרציה של חלקי הלומדה בדיקת הלומדה	מדולריות, נכונות ושיטות בדיקה
12	הכנת תיק תיעוד והנחיות להצגת הלומדה	תיעוד של תכנה
13	הצגת לומדה	ביצוע ההצגה

טופס הערכה

הערכה של למידה בגישה זו מתייחסת הן להערכה של התוצר הסופי, כלומר הלומדה שיצרו התלמידים, והן לתהליך העבודה ותוצרי הביניים. בנוסף, הערכה מתייחסת לממדי הידע השונים המתוארים בטבלה מס. 3. כדי להתאים שיטת הערכה לגישת למידה זו נשתמש בהערכה חלופית ונחבר מחוון אנליטי מתאים. מחוון אנליטי הוא אוסף של הנחיות לאופן הערכה ומתן הניקוד. מחוון מכיל:

- ממדים (Dimensions): קריטריונים של ידע או כישורים. לדוגמה, הערכת כישורים של עבודת צוות במטלת ביצוע אפשר ותכלול את הממדים הבאים: הסתגלות (הכרה בבעיות ותגובות מתאימות), תיאום (ארגון פעילויות קבוצתיות על מנת לסיים את המשימה במועד), קבלת החלטות (שימוש במידע נתון לקבלת החלטות), קשר בינאישי (אינטראקציה של שיתוף פעולה), מנהיגות (מתן הכוונה לקבוצה), תקשורת (החלפת מידע בצורה בהירה ומדויקת בין חברי הקבוצה).
- נקודת ציון: לכל קריטריון יפורט תיאור של רמה מסוימת בביצוע התלמיד, המצופה מן התלמידים ברמת גיל, כיתה או התפתחות מסוימת. נקודות ציון בדרך כלל מיוצגות על ידי דגימות של עבודות תלמידים.
- סולם דירוג: ערכים מדורגים הניתנים לביצועי התלמיד בממדים שונים.

טבלה מס. 5 מתארת דוגמה למחוון אנליטי:

ממדים	3 נקודות	2 נקודות	1 נקודות	0 נקודות
ידע במדעי המחשב	שימוש נכון ויעיל במבני תכנות שפורט בטבלה מס. 3.	שימוש סביר אך לא תמיד יעיל במבני תכנות. לדוגמה – שימוש בהוראות if ללא קיבולת כאשר ניתן לבצע קיבולת.	שימוש לא מתאימים לביצוע פעולות נמנע מלהשתמש במבני בקרה ושגרות	לא בוצע
הערכת התוצר – התוכנית	עומד בדרישות ומתאים לאפיון המשימה	עומד ברוב הדרישות ומתאים ברוב המקרים לאפיון המשימה	לא עומד בדרישות במרבית המקרים ואינו מתאים לאפיון המשימה	לא עומד בדרישות כלל
תהליך	תיאור מדויק של תהליך	תיאור חלקי של תהליך	תהליך הפיתוח	לא התייחס כלל

פתרון	פיתוח מתייחס לניתוח ותכנון מתאימים	הפיתוח המתאים לתוצר	המתואר ברובו אינו מתאים למוצר .	לתהליך הפיתוח
כישורי פתרון בעיות	שימוש בידע חדש שהתלמיד רכש באופן עצמאי, תהליך קבלת החלטות בפיתוח משקף שיקולים מתאימים במדעי המחשב	שימוש בידע חדש שהתלמיד רכש אך לא באופן עצמאי	אין שימוש בידע חדש מעבר למה שנלמד בכיתה	השתמש בידע ברמה נמוכה מאוד, לא מתאים לרמה נדרשת בקורס
עבודה בקבוצה	הקבוצה השלימה את בניית הלומדה בהצלחה	התוצר של הקבוצה עובד ברובו	תוצר הקבוצה לא פועל	תוצר הקבוצה אינה גמורה

לסיכום: פיתוח קורס או חלק בגישה זו יש לה יתרונות רבים במדעי המחשב. עם זאת הכנת סביבת למידה וחומרי עזר דורשת משאבים וזמן רב (לפחות בגרסאות הראשונות של חומרי העזר). בנוסף, יש לקחת בחשבון שבהתחלה יש לספק ידע נוסף כמו בתהליך פיתוח תכנה וכן המשימות הן גדולות יותר ולכן הוראת הנושאים לעיתים תארך זמן רב יותר מאשר בגישה המסורתית.

פרק 3. מינימליזם

3.1 מבוא – מהו מינימליזם

מינימליזם היא תיאוריה שצמחה בתחילת שנות ה-1980 כמודל לתיעוד תוכנות ופיתוח מדריכים למשתמש (tutorials). באותם השנים פותחו מחשבים ביתיים – מחשבי ה-PC והם נכנסו לשימוש נרחב בעסקים קטנים ובבתים פרטיים, וכך יותר ויותר אנשים ללא רקע במחשבים החלו להשתמש במחשב. כמו כן פותחו יישומים לשימוש כללי כמו מעבדי תמלילים וגיליונות אלקטרוניים שלוו במדריכים למשתמש. אולם, מה מאוד התברר כי המדריכים למשתמש שפותחו ליישומים אלו לא התאימו לצרכים של המשתמש הפשוט והשימוש ביישומים אלו (ובמחשב עצמו) גרם לתסכול רב בין המשתמשים. לכן חברת IBM הקימה צוות של חוקרים שתפקידם היה לחקור את האופן השימוש ביישומים שונים. צוות החוקרים ובראשם John M. Carroll (1998), ערך מחקר אמפירי שעקב ותעד את הפעולות שבצעו משתמשי מחשב בעבודתם עם יישומים, את המטרות שלהם ואת הרציונל שעליו בסו המשתמשים את הפעולות שלהם. החוקרים תיארו את המאבק של המשתמשים המנסים להפעיל את היישום כמיטב יכולתם ונכשלים פעם אחר פעם במשימתם. מחקרם שפך אור על תהליך למידה של משתמשי מחשב. המסקנות העיקריות ממחקרם הם:

- משתמשי מחשב מעדיפים "לעבוד מול המחשב" והם חסרי סבלנות לקריאת ההוראות (המדריך למשתמש). כאשר למשתמש חסר ידע מסוים הדרוש לביצוע המשימה, הוא מנסה לאלתר תוך שימוש בידע וניסיון קודמים כדי להמשיך הלאה בביצוע המשימה.
- במקרה של תקלה, המשתמשים בצעו פעולות שוב ושוב ולא ניסו ללמוד מניסיונם (ניסוי וטעייה). כלומר, הם לא ניסו לבצע ניסיונות שיטתיים ולהבין ולהסיק מסקנות על סמך ידע קודם כדי להבין מה קורה ולהתקדם הלאה.
- במקרים רבים חומרי התמיכה שליוו את התוכנות היו לא מתאימים. למרות שהיה תיעוד רב, הוא התמקד בתיאור המערכת, ולכן שהמשתמשים, שהתמקדו בביצוע משימה, לא השתמשו בו (הוא לא עזר להם לביצוע המשימה).
- בנוסף חומרי ההדרכה לא כללו תמיכה בזיהוי ואבחון תקלות (שטירון שלומד מערכת לא יכול להימנע מלבצע אותם) ובכלים להתגבר על תקלות אלה.

מכאן נגזרו עקרונות של הגישה של העיצוב המינימליסטי לתיעוד ומדריכי משתמש שבאמצעותם יכול משתמש ללמוד באופן עצמאי להשתמש בתוכנה. ניתן לסכם שלושה עקרונות מנחים בגישה זו:

- **למידה תוך כדי עשייה** ולא למידה על ידי קריאה (למידה מכוונת פעילויות).
- העסקת לומד בביצוע **משימה אמיתית** מוקדם ככל האפשר מעודד מוטיבציה ללמידה.
- **מתן תמיכה ללומד לזיהוי ואבחון תקלות** נפוצות והתאוששות מהירה.

למינימליזם יש הרבה משותף עם הקונסטרוקטיביזם. שתי התיאוריות הללו מתייחסות ללמידה אקטיבית המאפשרת ללומד להבנות את הידע של עצמו תוך כדי התנסויות ועל בסיס ידע קודם. החשיבות של המינימליזם הוא בכך שהוא מדגיש את השילוב של ידע ומיומנויות לזיהוי ואבחון תקלות כחלק מיחידת הלימוד ולא כסימפטום לכישלון או תפישה מוטעית של הלומד. אולם בניגוד לקונסטרוקטיביזם, מינימליזם דוגל בהוצאת כל מידע שאינו קשור ישירות לביצוע המשימה, או לפחות דחיית מתן אותו מידע לשלב מאוחר יותר. מודל זה לטענת המבקרים יכול להפריע ללומד, משום שלא בסיס ומודל מנטלי חיוני ומתאים למידה כזו יכולה להביא לומד למסקנות מוטעות ופיתוח מודל מנטלי לא מתאים.

בהתאם לתיאוריה זו ניתן לסכם עקרונות לתכנון חומרי למידה בגישה המינימליסטית:

- שימוש במשימות אמיתיות.
- תכנון סביבה המאפשרת לתלמידים תרגול של משימות מהר ככל האפשר.
- שילוב פעילות הכוללת חקירה וגילוי.
- הסתמכות על היכולת של לומד לבצע היסק ואימפרוביזציה על סמך ידע קודם.
- פיתוח חומרי תרגול כך שהם כוללים רק חומר רלוונטי לביצוע המשימה.
- פיתוח משימות שאינן כוללות ידע שלם ומתן הזדמנות ללומד להשלים ידע וביצוע עצמי של המשימות.
- תכנון חומר לימוד באופן מודולארי כך שיהיה ניתן לקרוא אותו בסדר שונה.
- תכנון חומר עזר המסייע בזיהוי ואבחון תקלות ומספק שיטות להתאוששות.
- שימוש בידע קודם של לומד, מוטיבציה ומטרותיו להכנת חומרי למידה מתאימים להם.

השפעת הטכנולוגיה על למידה

Carroll and Rosson (1987) פרסמו מאמר על הפרדוקס של פעילות משתמש (Paradox of the active use). לטענתם, שינויים בסביבה טכנולוגית של האנשים ובעיקר השימוש במחשב ככלי מתוחכם ורב עצמה ללמידה ועיבוד מידע, משפיע על הקוגניציה של משתמשי המחשב. במאמר זה מתארים החוקרים את שני פרדוקסים הנובעים מהתנגשות בין הקוגניציה (החשיבה) והמוטיבציה של משתמשים במחשב.

פרדוקס ראשון - "production bias", קיים אצל אנשים בעלי נטייה להתעסקות עם מוצרים ומטרתם העיקרית היא תפוקה. אנשים אלו נוטים להתמקד בביצוע פעולות עם המערכת וזה משפר את הסבירות שהם יעזרו בחומרים שונים שיסייעו להם בביצוע המשימה. אבל מצד שני, העדפה של פעולות ביצוע ותפעול מקטינה את המוטיבציה שלהם והם משקיעים משאבים רק כדי ללמוד על המערכת. לכן כאשר הם נתקלים במצבים הם נוטים להיצמד לפעולות שהם כבר מכירים, ואינם מנסים ללמוד דרך חדשה ויעילה יותר.

פרדוקס שני - "assimilation bias", הוא הנטייה של אנשים להטמעה ויישום ידע קודם כדי לפרש מצב חדש. נטייה זו יכולה לעזור בביצוע משימה חדשה, אבל מצד שני השימוש בידע קיים שאינו רלוונטי ומטעה במצב חדש, יכול להוביל את הלומד להסיק מסקנות מוטעות ולא להכיר באפשרויות חדשות מתאימות יותר.

לסיכום, Carroll and Rosson טוענים כי אנשים רוצים ללמוד תוך כדי עשייה, אבל גישה זו מטה אותם לקפוץ מסביב בגישה סתגלנית ברצף הלמידה. המשתמשים רוצים להבין דברים ולבנות את הבנתם לבד, אבל הם לא סבלנים ולעיתים הם מסיקים מסקנות שגויות. הם מנסים להשתמש ולהרחיב את הידע והכישורים הקודמים שלהם, אבל זה יכול להוביל להפרעות והכללות יתר. הם מנסים ללמוד תוך כדי אבחון תקלות והתאוששות, אבל שגיאות יכולות להיות מכשול ולהבנה וקשיים יכולים לגרום מהמוטיבציה. בהתאם ניסחו החוקרים חמישה כללי אצבע המתארים את ה-trade-off בין המאפיינים של המינימליזם לבין הקושי בתהליך למידה אותם מפתחי חומרי למידה צריכים לקחת בחשבון:

- עבודה על משימה מעשית מספקת ללומד מסגרת מתאימה לאינטגרציה וליישום של התנסויות למידה, אבל משימות אמיתיות יכולות להיות קשות מדי ולא יוכלו לאפשר תכנון תמיכה מתאימה.
- עבודה על משימה מוכרת מנחה ומניע לומדים להשתמש בידע קודם, אבל אינו מעודד למידה של המשימה הנוכחית, ויכול לגרום לשימוש לא מתאים בידע קודם.
- שילוב של תכנון וביצוע במשך התנסויות הלמידה עוזר לכונן את הלומד ליישם ידע ותומך בהעברה של כישורים, אבל מקטין את הסיבוכיות של המשימה.
- תרגול הכולל אחזור, ליטוש והסקת מסקנות של ידע וכישורים תומך ומעסיק את הלומד ומאפשר לו לרכוש כישורים יציבים יותר. אבל יכול שלא יהיה ללומד מספיק גישה למידע להיסק מוצלח, כמו כן הוא יכול להיות מודאג מאחריות על ביצוע משימה מורכבת כזו.

- אבחון והתאוששות מתקלות ממקד ומניע לומד ועוזר לו לחדד מושגים של ביצועים תקינים, אבל שגיאות יכול להיות מתסכלות ולהפריע בהשגת מטרות המשימה.

כדי לענות לצרכים אלו, מציעים החוקרים מספר אסטרטגיות כדי להתמודד עם הבעיות שנובעות מפרדוקסים אלו. א. כדי לנצל את היתרונות של התמקדות של הלומד במוצר הסופי על חשבו רכישת ידע ניתן:

- לבנות סביבת למידה שהיא תומכת בתגמול פנימי.
- להבטיח שהידע הוא פשוט וקל לאיתור
- לנצל את הרצון של לומד לבניית סביבה המניעה למידה
- ב. כדי לנצל את הנטייה של לומד ליישם שימוש בידע קודם ניתן:
 - מצד אחד למזער קשר הלא רצוי לידע קודם
 - לנצל מצבים בהם ניתן לנצל ידע קודם ביעילות
 - לנצל את ההתאמה בין מצבים במקרים שבהם ההטמעה נכשלת.

3.2 יישום מינימליזם במדעי המחשב

ככלל ניתן להשתמש בתיאוריה זו בכל תחום בהוראה, אולם מינימליזם התפתח בתחום הוראת מדעי המחשב, שבו חלק מהלימוד כרוך בהכרה ושימוש בסביבות ממוחשבות. בהמשך, בשנת 1998 כתב Carroll ספר חדש המרחיב את התיאוריה המינימליסטית ומתאר השימוש בה גם לעיצוב ופיתוח מערכת ממוחשבת (הנדסת תכנה) ולהוראת תכנות מונחה עצמים.

עיצוב מערכת ממוחשבת: Carroll הוא חקר ואפיין דרכים וגישות לפיתוח תכנות. למרות שהגישה הרווחת לפיתוח תכנות שנלמדה בשנות ה-1980 הייתה הגישה המובנית (top down), ראה Carroll שבמציאות מפתחים עובדים אחרת. באופן מעשי, טען Carroll, מהנדסי תכנה אינם מתכננים מערכת מתחילתה ועד סופה, אלא תוך כדי תהליך פיתוח הם בונים פתרונות ביניים ופתרונות חלקיים שמשמשים לבניה של האיטרציה הבאה של המערכת. כמו כן, לטענתו בגישה המובנית היה קושי בבניית המערכת, משום שהגדרת המערכת כולה נעשה בשלב הראשון, השלב היחיד בו המשתמש מעורב, ובסיום מקבל המשתמש מוצר מוגמר שקשה להכניס בו שינויים. לעומת, זאת היישומים העסיקים מטבעם הם דינמיים ודורשים הכנסת שינויים. אבחנות אלו הובילו את Carroll להגדיר מה שקרוי פיתוח איטראטיבי או פיתוח בגישה איטראטיבית, המערכת אינה מוגדרת בשלב הראשון כסופית, אלא מפותחת בתהליך איטראטיבי תוך כדי מפגש עם המשתמש ומתן ביטוי לדרישות ולהערות שלו שיורחבו למערכת מעוצבת המתאימה לצרכים של המשתמש.

אולם בשנות ה-1990 מערכות תכנה שפותחו הפכו להיות גדולות ומורכבות, והן מכילות אלפי (אם לא יותר) פעולות, ישויות, קשרים ואירועים ולכך גם גישת האב-טיפוס לא מתאימה. כפתרון לבעיה, חקר Carroll תיאוריות פסיכולוגיות על עיצוב בתחום מדעי – psychological design rational. הוא הציע להשתמש בתיאוריה המבוססת על היפותזה-היסק לוגי, לפיה לוקחים בחשבון תוצאות שהושגו בתצפיות וניסויים ישירים, ובאמצעות היסק מעלים היפותזה אותה מנסים לאמת בצורה לוגית ע"י נתונים שנגזרו מהניסויים.

3.3 מינימליזם ותכנות מונחה עצמים

ב-1988 החל Carroll לחקור סביבות מונחות עצמים ושפת התכנות Smalltalk. Carroll התעניין בבעיה שהציקה לרבים: איך לסייע למתכנתים מנוסים בפרדיגמה פרוצדורלית לעבור ולהפנים את עקרונות עיצוב מונחה עצמים. הבעיה העיקרית ביישום הגישה המינימליסטית בתכנות מונחה עצמים נבעה מכך ששמימה אמיתית היא גדולה ומורכבת מדי, ולכן קשה להציגה לטירון בצורה יעילה. כמו כן, ביצוע משימות גדולות מדי יכולות לגרום לתסכול רב. קושי אחר נבע מכמות הגדולה של מושגים שיש ללמוד בשלב התחלתי. קושי זה גרם לכך שבהוראה מסורתית,

פיתוח ועיצוב מערכת מונחת עצמים נדחה לשלבים מאוחרים של הקורס. במאמר שפורס על ידי Rosson ו-Carroll (1996) הם מתארים פיתוח של קורס לעיצוב מונחה עצמים OOD בגישה המינימליסטית, בו התלמידים מועסקים להתחלה בביצוע משימות אמיתיות, תוך כדי שימוש בתרחישים של משימות משתמש. ניתוח תרחישי משתמש אלו מסייעים ללומד לבנות מודל של עצם, ובהמשך מודלים אלו מוכללים ליצירת מחלקה ובסוף לעיצוב השלם של התכנה המורכבת מאירועים ותרחישי משתמש. המשימות תוכננו כך, שהן לא דורשות ידע קודם רב כדי להבין את המושגים ב-OO ולביצוען השתמשו התלמידים בספריה של מחלקות שהוספה לסביבת העבודה. המשימות הללו משמשות כמנוף ומסייעות לתלמיד להכיר שיטות של תכנון ועיצוב תוך כדי שימוש וניתוח תרחישים המבוססים על אינטראקציות עם המשתמש. במהלך הקורס, התלמידים לומדים לבנות מודלים קונקרטיים של תרחישים, ולנתחם בקונטקסט של המשימה וכך לגלות את היעילות של הפשטה ועקרונות OO. סביבת הפיתוח שבה השתמשו החוקרים כוללת מחולל תרחישים (Scenario Browser with Smalltalk).

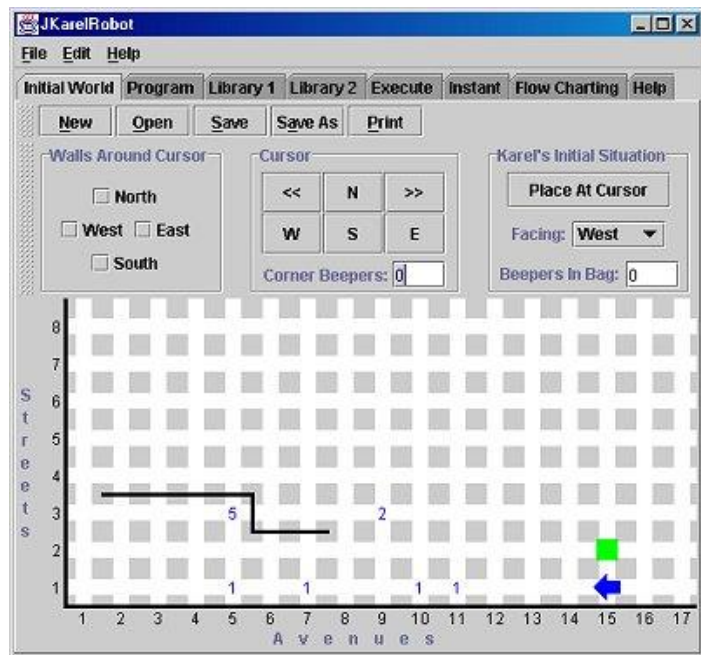
תיאור הפעילויות ומבנה הקורס:

- הנחיות ראשוניות: הקורס מתחיל בתיאור קצר ומונחה של הנושאים ב- OOD ובשיטת ההוראה והחומרים שעומדים לרשותם, ולאחר מכן מוצגים מספר נושאים בסיסיים בעיצוב. בכל נושא השתמשו התלמידים בחומרי עזר הכוללים תיאור אופן הביצוע פעילות עיצוב ספציפית והתנסות עצמית על פעילות דומה.
- תכנון תרחישי משימה: תרחישים אלו מתארים משתמש שעסוק בביצוע פעילות משמעותית עם מערכת אמיתית או דמיונית. כייצוג לעיצוב, התרחישים משקפים מושגים וערכים שהם נפוצים בין הרבה אנשי מקצוע ב-Human Computer Interface. התלמידים עוסקים בניתוח התרחישים שהם מוכרים להם כמשתמשים וכך מעוגנת ההוראה בביצוע פעילויות מוכרות וקונקרטיות. דוגמאות בהן השתמשו בקורס כדי להציג תרחישים הם: חנות למוצרי רכב, משחק בלקג'ק, איתור ספרים בספריה וכדומה. בשלב זה הניתוח מתמקד בהבנה וניתוח של ממשק משתמש ברמה בסיסית, כך שתלמיד יכול להבין את הפעילות שצריכה להתבצע והן את הרציונל של עיצוב של הממשק.
- זיהוי אובייקטים עיקריים: ניתוח מספר תרחישים משמש לבניית מודל של אובייקט. כדי לסייע לתלמיד להפעיל היוריסטיקה לזיהוי, הוצגה אסטרטגיה לפיה ניתן ל-"סמן את שמות עצם".
- קביעת תחום האחריות של אובייקט (Assigning Object Responsibility): המשימה שניתנה לתלמידים היא "להאניש" (anthropomorphize) את האובייקט שבתרחיש. במילים אחרות התלמידים משחקים את התפקיד של האובייקט הנתון ותוך כדי כך מגדירים את האחריות והתפקידים שלו. פעולת האנשה של האובייקט משמשת כמקור מידע על הידע שאובייקט צריך להכיל (מצב פנימי) על הקשרים שלו עם אובייקטים אחרים ועל התנהגויות שהוא צריך לתמוך.
- בניה תסריט לתרחיש: תרומת התרחיש ל- OOD היא אספקת תוכן והקשר שאתו יכול המתכנן לחקרו ולנתח. זה מספק את המבט הפרוצדורלי. במשימות אלו נדרשו התלמידים לבנות תסריט פעולה ("אלגוריתם") לשיטת אובייקט.
- ניצול של ידע משותף למספר אובייקטים: ניתוח האובייקטים שנבנו על סמך תרחישים מאפשר לתלמיד למצוא את המשותף בין מספר מופעים. על האובייקטים שהגדירו התלמידים בשלבים הקודמים, הם בפעילות זו מתבקשים לזהות את המשותף (מבנה והתנהגות) ואת השונה (ערכים), ובהתאם ליצור מחלקה שמכילה את המידע המשותף. תהליך דומה של הכללה של מחלקות יוצר את המושג היררכיה וירשה.
- פעילות בה תלמידים מנתחים את תפקידו של אובייקט, מוביל לזיהוי סוג האובייקט. לדוגמה סוג אובייקט יכול להיות שתפקידו: לתאם, לבקר, לבנות וכדומה. זיהוי התפקיד של אובייקט מקל על התכנון של מבנה והפעולות שלו.
- פיתוח אובייקט מופשט – התרחישים מנתחים בשני תנאים: פירוק מחדש של אובייקטים שיש בהם חוסר עקביות וסילוק של אובייקטים שקיבלו הרבה תפקידים מכדי שיהיה ניתן לכמס אותם באובייקט מופשט אחד.
- תמיכה בשימוש חוזר ושימוש חוזר על ידי הורשה: שימוש במשימות הכוללות מישק למשתמש המאפשר שימוש באובייקטים מובנים בסביבה.
- לסיום בנית מערכת המבוססת על שימוש במחלקות מוכנות ואובייקטים ומחלקות חדשות שהתלמידים יצרו.

3.4 קרל הרובוט כסביבה ליישום מינימליזם בתכנות מונחה עצמים בקורס מבוא.

תכנות מונחה עצמים היא פרדיגמה תכנותית המקובלת כיום בקורסי מבוא שונים ובהתאם היא תהיה הפרדיגמה הנלמדת ביסודות 1 ו-2 וביחידה ובעיצוב תכנה. בחרנו להבא דוגמה לפעילות שמתאימה לגישה המינימליסטית והיא מתאימה להוראת תכנות מונחה עצמים בקורס מבוא. כדי ליישם את העקרונות של הגישה המינימליסטית, חפשנו סביבת למידה שהשימוש בה יאפשר לתלמיד ללמוד תוך כדי התנסויות, מתחילת הקורס וללא ידע מוקדם רב, ותוך כדי כך ההתנסות לרכוש את המושגים הבסיסיים בתכנות מונחה עצמים (OOP), מושגים כמו אובייקט, מחלקה, שיטות, הורשה, פולימורפיזם וכדומה. סביבת הלמידה שבחרנו היא קרל הרובוט ששודרגה לתכנות מונחה עצמים והיא נקראת karl++ (סביבה שחלק מהמורים הכיר שתחילת בשנות ה-1990). סביבה זו קלה מאוד להפעלה והיא מאפשרת לתלמיד לבצע משימות על רובוטים מוכנים מיד בתחילת הקורס. אבל יחד עם הפשטות שלה, היא מאפשרת להציג את עקרונות תכנות מונחה העצמים בשפת Java או C++.

הסביבה מבוססת על רובוט היכול לנוע בעולם שאנו יכולים ליצור עבורו (איור מס. 1), עולם הכולל מכשולים. הפעולות שאנו יכולים להורות לרובוט לבצע הם פעולות כמו להתקדם לכיוון מסוים, להסתובב, להניח זמזמים ולאסוף אותם מנקודות מוגדרות. שימוש בפעולות הבסיסיות והמובנות בסביבה, מאפשר לנו ליצור מאגר משימות ולאפשר לתלמיד לתרגל מיד בהתחלה. בנית העולם של הרובוט היא פשוטה מאוד ומתבצעת באמצעות כלים גרפיים.



איור מס. 2 – העולמות של Karl++

ההקבלה בין המושג רובוט (שהוא דבר ויזואלי המוכר ומובן לתלמיד) למושג אובייקט, היא פשוטה ואינטואיטיבית והיא מאפשרת הצגה של עולם המושגים בתכנות מונחה עצמים. לדוגמה: פעולות שהרובוט עושה הן שיטות (Methods). הוראה לרובוט לבצע פעולה מתקשרת למושג message. הפעלת הרובוט מציגה את המושג משימה ותוכנית. משימה הכוללת מספר רובוטים מתקשרת למושגים מחלקה ומופע. שימוש במשימה המרחיבה ומוסיפה פעולות חדשות לרובוט, לדוגמה: לך 10 צעדים במקום ללכת צעד בודד, מרחיבה את השיטות של הרובוט. המושג הורשה יכול להיות מוצג במשימה בה יש לבנות שני רובוטים שתפקידם להניח זמזמים, כאשר רובוט אחד מניח זמזם בכל צומת, ורובוט שני מניח זמזם נוסף בכל צומת שלישי בכל צומת זוגי. כדי לפתור את הבעיה נשתמש בשני רובוטים כשרובוט השני יורש פעולות מהרובוט הראשון ובנוסף מוטלות עליו פעולות ייחודיות. גם המושג פולימורפיזם מוצג די בשלב מוקדם של הקורס באמצעות בניית מספר רובוטים שכל אחד מהם מבצע משימה אחרת.

הצעה לתכנון חלק מקורס CS1 המשתמש בקרל הרובוט

ההצעה לקוחה מאתר של קרל הרובוט. משך הקורס הוא חמישה שבועות. אלו חמשת השבועות הראשונים בקורס מבוא במדעי המחשב ובו מוצגים המושגים: אובייקט, מחלקה, הורשה, בחירה, ביצוע איטראטיבי.

תיאור הקורס:

שבוע ראשון – **המושג אובייקט ומחלקה**: בשבוע ראשון לומדים התלמידים להתקין ולהשתמש בתכנה ומתוארים הרובוטים והעולם בהם הוא גרים. לאחר התנסות בביצוע מספר תכניות מוכנות וסקירה של קוד המקור, תלמידים יכולים לאתחל מופע של רובוט, לעורר ביצוע שיטה של רובוט, לבנות יישומן הכולל מספר רובוטים זהים (השייכים לאותה מחלקה), כשכל אחד מהם יש מצב התחלתי שונה משימה טיפוסית.

שבוע שני – הרחבת **שיטות של אובייקט**: בסיום השבוע הראשון היא ההקבלה בין המושג רובוט (שהוא דבר ויזואלי המוכר ומוכן לתלמיד) למושג אובייקט היא פשוטה ואינטואיטיבית. להתקדם צעד ימינה. בשבוע זה התלמידים לומדים כיצד לאפשר לרובוט לרכוש התנהגות חדשה, למשל התקדם 10 צעדים ימינה. רעיון זה ניתן להרחבה ומאפשר לבנות מחלקה חדשה ובה להשתמש כדי לפתור בעיות מורכבות יותר.

- שבוע שלישי - **מבני בקרה**: בשבוע זה ילמדו שיטות חדשות של הרובוט המחזירות ערכים בוליאניים, לדוגמה שיטה הבודקת אם יש קיר לפני הרובוט (frontIsClear) ומחזירה ערך בוליאני מתאים. שימוש בשיטות אלו מספק הזדמנות ללמד את מבני הבקרה וביצוע חוזר.

- שבוע רביעי - **שיטות והעברת פרמטרים**: בשבוע זה מוצגת שיטה לעברת פרמטרים כדי לפתור בעיה כמו: אסוף מספר (המועבר כפרמטר) של זמזמים. משימה זו מאפשרת להבין את השימוש בפרמטרים על פי ערך.

- שבוע חמישי- **משתני מופע**: בשבוע זה מורחב המושג משתנה של מופע. לדוגמה, ברצוננו שרובוט ישמור את מספר הזמזמים שאסף ולשם כך יש להשתמש במשתנה מתאים.

כפי שניתן לראות, השימוש בקרל הרובוט מאפשר:

- הצגה של המושגים הבסיסיים בתכנות מונחה עצמים מיד עם תחילת הקורס.
- הקורס מאפשר לתלמיד לבצע משימות מורכבות יחסית בתחילת הקורס.
- הקורס בנוי כך שהוא בונה כל נושא על בסיס המשימות והידע הקודם.
- סביבת הלמידה היא קלה ומאפשרת התנסות מהירה.
- בכל נושא מוצג המידע המינימלי הדרוש להשלמת המשימות.
- השימוש בתכנה הוא מהנה ומגביר את המוטיבציה ללמידה.

3.5 יישום הגישה המינימליסטית ביחידת הלימוד 'מבוא למערכות מידע'

נציג דוגמה נוספת לשימוש בגישה המינימליסטית במקצוע 'מבוא למערכות מידע'. מקצוע זה עוסק בתכנון ועיצוב של מסד נתונים טבלאי. הקורס כולל עקרונות ותיאוריה מבנה ותכנון של מסד נתונים טבלאי, תיאור של תהליך פיתוח מערכת מידע הממומשת באמצעות מסד נתונים טבלאי ושימוש במחולל יישומים, לדוגמה Access, כסביבת עבודה. בחרנו להדגים פיתוח חומרי למידה להוראת מקצוע זה בגישה המינימליסטית, משום שהמקצוע 'מבוא למערכות מידע' עוסק בהיבט יישומי של מדעי המחשב – בנייה ושימוש במערכת מידע. בסעיף זה, נתאר תחילה המלצות כלליות להוראת המקצוע ואחר נתאר דוגמה לפעילות.

סדר ההוראה:

בהתאם לגישה מינימליסטית ארגון סדר הלמידה צריך לאפשר ללומד להתחיל מוקדם ככל אפשר בהתנסות, ולכן מומלץ להתחיל מהסוף להתחלה, כלומר משימוש במוצר המוגמר אל לתהליך הפיתוח. בהתאם לכך הצגת נושאי הלימוד יתבצע בסדר הבא:

- שיעור מבוא קצר המתאר את היישומים והשימושים במסדי נתונים טבלאיים (כפי שהם באים לידי ביטוי במערכות מוכרות לתלמידים כמו ספריה, ניהול טיסות, מערכת ציונים בבית ספר, מנועי חיפוש ועוד).
- תרגול המאפשר תפעול של **מסד נתונים טבלאי קיים** (הוכן מראש עבור התלמידים). התרגול מתייחס למסד נתונים טבלאי הכולל שתי טבלאות והוא כולל: פעולות עדכון (הוספה, שינוי וביטול) של רשומות בודדות ושל קבוצה של רשומות, הפקת מידע מהמסד באמצעות שאילתות, אחזור רשומה בודדת ואחזור קבוצה של רשומות, הפקת דוחות מסכמים. כמו כן הוא מאפשר שליפת מידע משולב משתי הטבלאות. תרגול זה מתייחס לנושאי הלימוד הבאים: מושג הטבלה, תפעול (עדכון והפקת מידע) טבלה וקשר בין טבלאות במסד (בהתאם לפרקים 2 ו-3 בתוכנית הלימודים: פרק 2: מאגרי מידע I – הטבלה ופרק 3: מאגרי נתונים II - מאגר נתונים טבלאי בתוכנית הלימודים)
- לאחר ההתנסות בתפעול של מסד נתונים טבלאי והכרות עם המושגים הבסיסיים, נסכם את התפקידים והשירותים של מערכת לניהול מסדי נתונים (פרק 4) ונתאר אופן בניית מסד נתונים שאפיונו מוגדר על ידי המורה שמשמש הן כמנתח מערכות והן כמשתמש. (בניית מסד נתונים טבלאי מתייחס לנושאים בפרקים 2 ו-3 בתוכנית הלימודים ולפרק 8 – ממשק).
- תכנון מסד נתונים טבלאי המציג את העקרונות והשיטות באמצעותן הוגדר מסד הנתונים הטבלאי אותו בנו בשלב הקודם (תכנון זה מתייחס לפרק 6 בתוכנית הלימודים).
- בשלב האחרון נתאר ונתרגל בניית מסד נתונים טבלאי לארגון ונתייחס לצרכים של ארגון ולתהליך הניתוח (פרקים 1, 6 ו-7 בתוכנית הלימודים)
- פרויקט מסכם המדגים את התהליך האמיתי של הפיתוח והבניה ומתחיל בשלב הגדרת מערכת המידע, ניתוח צרכים, תכנון מסד ובנייתו.

ידע קודם של תלמיד:

- מסד הנתונים אותו מפעיל התלמיד בשלב הראשון צריך להיות קשור לתחום שתלמיד מכיר היטב, כך נוכל להימנע משימוש ומתן מידע נוסף (הדרוש כדי להבין את התחום). לדוגמה, שימוש במסד נתונים המציג סרטים, ספרים, קבוצות כדורגל וכדומה, הוא נושא הקרוב לתלמיד והוא מכיר אותם ולכן כמשתמש במערכות מסוג זה, הוא יכול להבין את מטרות השימוש במסד נתונים. לעומת זאת, מסד נתונים המטפל במלאי מתייחס למושגים ופעולות שאינם מוכרים לתלמידים ומצריך מתן רקע במושגי יסוד של המערכת (מלאי, תעודת משלוח, ...).
- הפעלת מסד נתונים טבלאי דורש הכנת Tutorial. השתמשו בידע קודם של תלמיד בתפעול יישומי מחשב והימנעו מלהרחיב בהוראות שהן ברורות (התלמיד יאלתר ויקשר לניסיון קודם שרכש) ובהתאם הרחיבו בהסברים בפעולות שאינן מוכרות.
- בתיאור של הפעילות יש לשים דגש על הרציונל, כלומר יש לתאר את המטרות המושגות בביצוע משימה ולא להתייחס רק לתיאור הפעולות עצמן. לדוגמה, ביצוע פעולות אחזור תכיל הסבר על אפשרויות השימוש באחזור בצד הוראות ביצוע על מסד הנתונים הספציפי אותו מתרגלים התלמידים. ההוראות האלה צריכות לספק לתלמיד מספיק מידע כדי לרכוש יכולת לתכנן ולהפעיל מסד נתונים טבלאי אחר.

התייחסות לאבחון והתאוששות מתקלות

- ב- Tutorial המלווה את מסד הנתונים המוכן אותו מתרגלים התלמידים, רצוי להתייחס לשני סוגים של תקלות:
- סוג ראשון הן תקלות הנגרמות מהפעלה לא מיומנת של התכנה, לדוגמה הקובץ לא נמצא בספרייה הנכונה, פעולת שמירה של מסד נתונים מעודכן שלא הצליחה, הוספת רשומה חדשה שלא מצליחה משום שקיים כבר מפתח זהה ברשומה אחרת.
 - סוג שני מתייחס לתקלות שמשתמש יכול להיתקל בהן והן נובעות מחוסר תשומת לב בתכנון מסד הנתונים. לדוגמה, חוסר בבדיקות קלט, מחסור בהודעות פלט במקרים מסוימים (למשל אחזור רשומה שנכשל ולא מלווה בהודעה מתאימה יכול לתת את הרושם שהפעולה לא בוצעה כלל).
- רצוי שהתרגול על מסד הנתונים המוכן יכיל פעילויות הקשורות לתקלות יזומות. הסברים על תקלות אפשריות והתנסות באבחון תקלות, מעורר את המודעות של התלמיד לתקלות אליהן הוא יצטרך להתייחס ולמנוע במערכת שהוא יפתח.

פעילות לדוגמה של תכנון ובנייה טבלה המכילה מידע על תקליטי מוזיקה:

תכנון טבלה של תקליטי מוזיקה היא חלק ממשימה מחיי היום-יום (מחשוב חנות מוזיקה), והוא גם תחום הלקוח מעולם התלמיד. התכנון משלב פעילות של תלמיד יחד עם הקניית ידע והצגת המושגים ותהליך של תכנון מסד נתונים לתקליטים. התכנון מתחיל מאפיון מבט של משתמש, שהוא המבט שקרוב לעולם התלמיד. לאחר שהתלמיד מתבונן בדוגמאות, הוא מגדיר את דרישות המערכת ולאחר מכן מתואר תהליך הנרמול והבניה של המסד, אותו מיישם התלמיד על המערכת שהוא בונה.

נתאר את שלבי הביצוע:

- א. מבוא בו מתוארת המשימה - בנית מסד נתונים טבלאי לאיתור וניהול תקליטי מוזיקה. כמו כן נתונים הנחיות כלליות המגדירות את הדרישות, תיאור חומרי עזר, לו"ז ושיטות עבודה (כמו עבודה בקבוצות, דוחות שיש להגיש וכדומה).
- ב. משימת ביצוע מס. 1: הגדרת דרישות לאחזור מידע ממסד הנתונים בנושא מוזיקה. התלמידים יתבקשו לרשום את השאלות שלדעתם מערכת כזו צריכה לבצע. הם יכולים להיעזר במערכת קיימת בחנות מוזיקה קרובה למקום מגוריהם או להשתמש באינטרנט ולבקר בחנויות יש להם אתרים וירטואליים (לדוגמה האוזן השלישית, מיתוס וכדומה).
- ג. פעילות בכיתה: דיון בדרישות שהגדירו התלמידים, בניית מאגר של סוגי פעולות ושאלות כמו שאלות אחזור, עדכון, סיכומים וכדומה.
- ד. משימת ביצוע מס. 2: על התלמיד להגדיר את סוגי הפעולות והשאלות של המאגר שהוא בונה והתאם לחפש ולאסוף נתונים על תקליטים שיוכנסו לטבלה. (מקורות הידע בהם יוכל להשתמש אם המקורות שצוינו במשימה 1).
- ה. פעילות בכיתה: דיון בסוגי נתונים ומידת התאמתם למסד נתונים שיאפשר ביצוע השאלות. משימת ביצוע מס. 3: תלמיד יגדיר ויבנה טבלה מתאימה לתוכה הוא יכניס את הנתונים שנאספו במשימת ביצוע מס. 2.
- ו. פעילות בכיתה: תיאור הדרך למימוש פעולות אחזור מידע. רצוי להדגים ולהשתמש בחנות וירטואלית קיימת כדי להראות הכנסת ערכים, ביצוע חתכים וכדומה. שימוש בדוגמה מוכנה תאפשר להדגים גם את הצורך במסגרות קלט ופלטים מתאימים המטפלים בפעולות שגויות של משתמש.
- ז. משימת ביצוע מס. 4: בניית השאלות.
- ח. משימת ביצוע מס. 5: הערכה של הטבלאות. תלמידים יבצעו הערכת עמיתים וכל קבוצה תבדוק את המסד של קבוצה אחרת.

פעילות זו תוכננה בהתאם לגישה המינימליסטית והיא מאפשרת לתלמיד: להיות פעיל ואקטיבי מתחילת הקורס, להשתמש בידע ובניסיון קודמים שלו, ולהשתמש בכל שלב במידע הרלוונטי לאותו שלב (ומה שלא נחוץ נדחה לשלב מאוחר יותר). דוגמה נוספת לפעילות שמתוכננת בגישה המינימליסטית ניתן למצוא באתר מחשבה, בקובץ שנמצא ביחידת הלימוד 'מבוא למערכות מידע' ותחת הנושא פרויקטים. הקובץ מכיל תיאור של פעילות הקרוי 'אירוע מתפתח - ספריה', באמצעותו תלמיד לומד לאפיין, לתכנן ולבנות מסד נתונים לספריה. קובץ זה מכיל הנחיות למורה הכוללות מערך שיעורים מומלץ לביצוע משימה זו בכיתה.

פרק 4. תמצית המאמרים

נביא כעת סיכום תמציתי של 6 מאמרים מובילים בתחומים שסקר הספרות עוסק בהם.

מאמר ראשון:

ברוקס ז'. ג., וברוקס מ. ג' (1997). לקראת הוראה קונסטרוקטיביסטית : בחיפוש אחר הבנה. ירושלים: ת"ל ומכון ברקנו וייס.

מאמר שני:

Ben-Ari. M., (2001). Constructivism in computer science education. *Journal of Computers in Mathematics & Science Teaching*, 20(1):45—73.

מאמר שלישי:

Carroll, J.M. (1998). Reconstructing Minimalist. from his book *Minimalism: Beyond the Nurnberg Funnel* and reprinted by permission of the publisher (MIT Press).

מאמר רביעי:

Rosson, M.B. and Carroll, J.M. (1996). *Object-Oriented Design from User Scenarios*. Conference on Human Factors in Computing Systems, ACM Press New York, NY, USA

מאמר חמישי:

Savery, J. R., and Duffy, T. M. (1995). *Problem based learning: An instructional model and its constructivist framework*. *Educational Technology*, v 35, Sep-Oct. pp. 31-38.

מאמר שישי:

Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J.H. and Crawford, K., (2000). *Problem-Based Learning for Foundation Computer Science Courses*. *Computer Science Education*.

לקראת הוראה קונסטרוקטיביסטית : בחיפוש אחר הבנה. ברוקס ז'. ג. , וברוקס מ. ג' (1997).

ברוקס וברוקס טוענים כי ישום פדגוגיה בגישה הקונסטרוקטיביסטית מציבה אתגרים לא מעטים בפני מחנכים. לפי הגישה הקונסטרוקטיביסטית למידה היא תהליך של בניה של הבנות על העולם האמיתי. בית ספר הוא מתווך בין העולם האמיתי ללומד ומהיותו כזה הוא אינו יכול לספק את כל העושר והמורכבות של העולם. בנוסף, הלמידה מתבצעת באמצעות התיווך של מורים וחומרי למידה. ההוראה כמתווכת יוצרת עולם מושגים ותופעות שהן ייצוג של העולם אותו מנסים להעביר לתלמידים, אשר הם בתורם יוצרים לעצמם את הייצוג שלהם. בעיה נוספת נובעת מכך שבית ספר בוחר את הנושאים והמידע שתלמיד ילמד, ובכך מרחיק את הלומד מהקונטקסט. כלומר הידע שנלמד בסביבה די סינתטית ולא בסביבה האמיתית. בית ספר מנסה ליצור אחידות הן בנושאים הנלמדים והן בצורה הנלמדת וזאת בניגוד לגישה הקונסטרוקטיביסטית המדגישה את המגוונות והסובייקטיביות של הידע אצל הלומד. קושי נוסף כרוך בסביבת הלמידה בבית הספר בו עומד הקוריקולום הקובע את הנושאים, מפרק את הנושאים למושגים וקובע סדר הצגתם. בנוסף המורה במרכז והוא בעל הידע, שקובע מה חשוב ומה לא והוא מוביל את תהליך הלמידה. סביבת למידה כזו הופכת את הלומד לפחות אקטיבי ויותר פסיבי בלמידתו.

בהתאם, מציעים הכותבים חמישה עקרונות ליישום פדגוגיה בגישה קונסטרוקטיביסטית:

1. העלאת בעיות בעלות רלוונטיות ראשוניות לתלמידים- אין זה אומר שכל נושא נלמד רלוונטי מראש לכל התלמידים, אלא המורה מתווך ועוזר בבניית ההבנות על חשיבותם של הנושאים הנלמדים .
 2. הבניית הלמידה סביב "רעיונות גדולים" או מושגי יסוד - תכנית הלימודים מוצגת "מן השלם אל החלק", תוך הדגשת המושגים הרחבים. זאת מתוך הנחה שטבען של השאלות המוצגות לתלמידים משפיע במידה רבה על עומק החיפוש של התלמידים אחר תשובות. שאלה רחבה מאפשרת לתלמידים נקודות כניסה מרובות. היא מזמינה כל תלמיד להשתתף בפתרון, בדרכו הייחודית. מכאן מובן שכל נושא נלמד הוא רב תחומי כי רק בכזה מתקיימת חקירה אמיתית.
 3. ביקוש וייחוס ערך לנקודות מבט של תלמידים - דרישה זו קשורה ישירות לפילוסופיה של הקונסטרוקטיביזם הטוענת שאין אמת מוחלטת וכל דבר הוא יחסי. על כל דבר אפשר להסתכל מנקודות ראות שונות ולהבין אותו באופן שונה. המורה הקונסטרוקטיביסטי צריך להכיר ולהוקיר את נקודות הראות השונות של תלמידיו ולהעריכן.
 4. התאמת תוכניות לימודים כך שתתייחסנה לידע המוקדם של תלמידים- בסעיף זה הכוונה היא שתוכנית הלימודים והמורה יתחשבו ביכולות שהתלמיד מביא עמו לכתה - יכולות קוגניטיביות, חברתיות ורגשיות.
 5. הערכת הלמידה בהקשר הלימודי- הערכת הלמידה צריכה להיות בהקשר משמעותי לתלמיד ולהתייחס לבעיות שהתלמידים נתקלים בהן תוך כדי למידה. אסור לה להיות שיפוטית. מבחנים לסוגיהם אינם מתאימים לגישה הקונסטרוקטיביסטית. למעשה ההערכה בגישה זו היא חלק מההוראה: באמצעות השתתפות בפעילויות גומלין של תלמידים ומורים, באמצעות תצפית בפעילויות גומלין של תלמידים, ובאמצעות התבוננות בתלמידים עובדים עם רעיונות וחומרים. הלמידה ממשיכה גם בזמן ההערכה.
- פוסט מוסיפה את ההפשטה הרפלקטיבית כאחד העקרונות של הפדגוגיה הקונסטרוקטיביסטית. לדעתה זהו הכוח המניע את הלמידה, כיוון שבני אדם שואפים לארגן ולהכליל את התנסויותיהם בצורת ייצוג מופשט. מתן זמן לחשיבה, ייצוג בצורה רב-סמלית ודין בקשרים בין התנסויות, עשויים לדעתה לאפשר הפשטה רפלקטיבית (Fosnot, 1996).

לסיכום, בסביבה קונסטרוקטיביסטית הכיתה היא קהילת לומדים המעורבת בפעילות, שיח וחיבה משותפת, על אודות נושאים משמעותיים ללומדים. המורים מעודדים ומקבלים אוטונומיה של תלמידים, ויוצרים אווירה נוחה להתבטאות. תשומת לב רבה ניתנת ללומד: לידע הקודם שלו, לניסונו, לאסטרטגיות החשיבה שלו ולמוטיבציה. בהתאם לכל אלה מנווט המורה את הנחייתו לתלמיד. חשוב שסביבת הלמידה תהיה עשירה ככל האפשר ורב תחומית. חומרי גלם ראשוניים משמשים לצרכי מחקר. חשיבתם של התלמידים היא המניעה את השיעורים ודיאלוג, חקר והתלבטות זוכים להערכה. סביבת למידה קונסטרוקטיביסטית היא סביבה שבה התלמידים יחפשו משמעות, עריכו אי-ודאות ויגלו מהי אחריות.

Constructivism in computer science education. Ben-Ari. M., (2001).

קונסטרוקטיביזם היא תיאוריה של למידה לפיה, תלמיד מבנה את הידע של עצמו על בסיס התנסויותיו וידע קודם. ידע נבנה בצורה מתמדת בתהליך ריקורסיבי בו, ידע חדש נבנה ומוטמע על בסיס ידע קודם, ועל בסיס ידע זה נוצר ידע חדש. במאמר שני חלקים: חלק ראשון סקירה של התיאוריה הקונסטרוקטיביסטית והיישום שלה בחינוך מדעי ובחלק השני אנליזה של התיאוריה הקונסטרוקטיביסטית והשלכה על הוראת מדעי המחשב.

בן-ארי טוען כי בהוראת מדעי המחשב אנו נתקלים בשתי בעיות שהן ייחודיות למקצוע זה. בעיה ראשונה נובעת מכך שמדעי המחשב עוסק במוצרים מלאכותיים (artifact) – שפות תכנות ותוכנות, להן היוצרים בנו מודלים מסוימים והלומד חייב לבנות מודלים דומים. בעיה שנייה היא שידע אינו ניתן למשא ומתן חברתי. התכנה או המחשב מבצעים את מה שהם הוגדרו לבצע וזו המציאות בה אנחנו עובדים. לכן הלומד חייב להכירה את המציאות כפי שהיא ואין מקום למשא ומתן או להציע פירושים ופתרונות אחרים.

את המסגרת החינוכית של הגישה הקונסטרוקטיביסטית ניתן לתאר באמצעות ארבעה מרכיבים:

- אונתולוגי - חקר ההוויה (המציאות). בקונסטרוקטיביזם אין אמת מוחלטת וקיימות תפישות רבות ושונות, לכן היא לא יכולה להשפיע על הפרדיגמה החינוכית.
- אפיסטמולוגיה - חקר הידיעה. ידע מוגדר על האינדיבידואל ולכן הוא לא בהכרח חיוני.
- ידע נבנה באופן ריקורסיבי על בסיס ידע והתנסויות קודמות.
- למידה היא אקטיבית וכל לומד מבנה את הידע על בסיס ידע קודם.

על פי הגישה הקונסטרוקטיביסטית, למידה אפקטיבית דורשת בניה של מודלים מנטאליים חיוניים (viable), ותפקיד המורה הוא להנחות את הלומד בצורה אקטיבית. המשימה של מורה במסגרת זו היא מורכבת מאוד משום שעליו להיות ער לעובדה שכל תלמיד מבנה את הידע שלו ולכל אחד יכול להיות מבנה שונה. מורה צריך לקחת בחשבון שתלמידים המשתמשים בספרים ושומעים הרצאות, יכולים לתת פירושים שונים לידע המועבר. למידת חקר וביצוע פרויקטים הם שיטות המתאימות לגישה הקונסטרוקטיביסטית, משום שהן מאפשרות לתלמיד להיות אקטיבי ולרכוש את הידע שלו בעצמו. לשיטות אלו יש יתרונות הנובעים מכך שהם:

- מגבירים את הפוטנציאל האינטלקטואלי
- מתמקדים בתגמול פנימי לעומת תגמול חיצוני
- מדגישים למידה של אסטרטגיות והיריסטיקות
- משמשים כאמצעי לשימור הזיכרון

מאחר ובגישה הקונסטרוקטיביסטית אין אמת אחת, לא ניתן לטעון שמבנה קוגניטיבי הוא שגוי. פון-גלזרפד מציע להשתמש במילה חיוני (viable), כלומר משהו שהוכח כמתאים בקונטקסט שהוא נוצר בו. לכן, תפישה שגויה נתפשת לא כשגיאה, אלא בנייה לוגית של מודל לא חיוני. הוראה מתאימה צריכה לחשוף את המודל המנטאלי שבבסיס התפישה השגויה, ולהציע דרך לבניה של מודל חיוני על בסיס ידע קודם.

בן-ארי טוען כי הוראת מדעי המחשב היא קשה ובמיוחד לתלמידים מתחילים. מחקרים אמפיריים בהוראת מדעי המחשב תיארו את קשיים שניתן להסבירם במסגרת הקונסטרוקטיביסטית, לדוגמה:

- מושג משתנה נתפס כמושג קשה. התלמיד מאמין שלמשתנה B אין ערך לאחר ביצוע הוראת השמה $A \leftarrow B$. תפישה שגויה זו נובעת משימוש במטפורה של משתנה כקופסה (ידע קודם).
- לתלמידים יש קשיים בהבנת הוראות קלט/פלט. לדוגמה ביצוע 2 הוראות קלט בזו אחר זו לאותו משתנה נתפס כבלתי אפשרי. כמו כן לא ברור לתלמיד מי אחראי על ביצוע ההוראה.
- קיימים קשיים בהבנת המושג פרמטר יחסי. תלמידים משתמשים בידע קודם ומפרשים אותו על פי מודל של תקשורת לצורך העברת מידע ולא במודל האמיתי שהוא העברה והעתקה.

במאמרו טוען בן ארי שבהוראת מדעי המחשב יש לקחת בחשבון שני גורמים חשובים המשפיעים על למידה:

1. לתלמיד מתחיל אין מודל מנטאלי אפקטיבי של מחשב שהוא חיוני לפרשנות ובניית ידע חדש על בסיס התנסויותיו בכיתה.

2. המחשב יוצר מציאות אונותולוגית נגישה וקל לקבל תשובה שיכולה להתפרש כ"נכונה" ולכן כדי להשיג ביצועים מוצלחים יש לבנות של מודל נורמטיבי של המציאות.

בן ארי מצרף מספר המלצות לתכנון הוראה ב-CS :

- יש ללמד מודל של מחשב המתייחס למבנה חומרה וזרימת מידע בו בצורה מפורשת. השאלה היא מה רמת פירוט הדרושה ולכך יש מספר תפישות החל מפירוט הרכיבים האלקטרוניים וכלה בוויזואליזציה של מושגים כמו העברת פרמטרים. ההיקף קשור כמובן למטרות ההוראה, והוא יבחר כך שהוא יתמוך בהבנה של תהליך ביצוע תכנית על המחשב.

- מודל חיוני של מחשב צריך להילמד לפני האבסטרקציה, גם בשפות מונחות עצמים (הרמה בה אנו מתכנתים בשפות אלו היא מופשטת ורחוקה מאוד מן המחשב שברמה התחתונה).

- אסור שהמציאות המפתה של המחשב תתפוס את המקום של בניית המודל המנטאלי.

- צריך להכיר בעובדה שיש גישות שונות לתכנות שנוגעות מתכנון מלמעלה למטה ועד בריקולז. בריקולז היא גישה לתכנות (שתוארה על ידי Turkel and Papert) הניתנת לתיאור מלמעלה למטה, או תכנות על ידי "מימוש" והתנסות. כלומר קודם יש התנסות עם העצמים ואז יש בניה של מושגים ועצמים חדשים. בן-ארי טוען שכולנו משתמשים ברמה כלשהי של בריקולז, אבל שימוש בבריקולז ללא מודל מנטאלי מתאים והבנה טובה של המושגים והפרקטיקה יכולה להפריע לתלמידים מתחילים.

- משימות מעבדה מאפשרות לתלמיד להיות אקטיבי ולבנות ידע על בסיס התנסות ולכן דרושה תשומת לב רבה לתכנון נכון של פעילויות מעבדה. לטענת בן-ארי, עדיף לשלב משימות סגורות וסוג המשימות צריך לעודד תהליכים קוגניטיביים כמו ניתוח, בחינת אפשרויות וקבלת החלטות.

- תשומת לב צריכה להינתן לשיטת הערכה. בניגוד למבחנים בהם מעריכים תוצאות, הערכה במדעי המחשב צריכה להיות לא רק לתוצר אלא בעיקר לתהליך של פתרון הבעיות.

לסיכום מציע בן-ארי מספר הנחיות מעשיות לפדגוגים:

- בלי קשר לשיטת ההוראה (הרצאה, משימה, מעבדה), על המורה להגדיר בצורה ברורה את השינוי הקוגניטיבי הרצוי מתהליך הלמידה ולתכנן בהתאם את הפעילויות שישגו מטרות אלה.

- על המורה "לחפור מתחת" לידע והמומחיות שלו ולחשוף ידע קודם הדרוש כדי לבנות מודל מנטאלי חיוני של הנושא שהוא רוצה ללמד, ובתאם עליו להיות בטוח שלתלמידיו יש את הידע הדרוש.

- בכל קורס בו יש התייחסות לרמה של אבסטרקציה, יש להציג במפורש את המודל המנטאלי ברמה נמוכה יותר שהיא בבסיסה של האבסטרקציה.

- כאשר תלמידים שוגים או אינם מבינים, יש להניח שלתלמידים יש מודל מנטאלי לא חיוני. המשימה של המורה היא לחשוף מודל זה ולסייע לו להתאים את המודל המנטאלי.

- על המורה לספק הזדמנויות רבות ככל האפשר לביצוע רפלקציה עצמית (לדוגמה על ידי ניתוח שגיאות) לאינטראקציה חברתית (לדוגמה עבודה בקבוצות).

Reconstructing Minimalist Carroll, J.M. (1998).

מאמר זה סוקר את התפתחות הגישה המינימליסטית כמסגרת לעיצוב הוראה ובעיקר מדריכים למשתמשי מחשב. התיאוריה מדגישה את הצורך בבניית ידע של ידי התנסויות וביצוע משימות. Carroll מתאר את עקרונות הגישה לפיתוח מדריכי משתמש ליישומי מחשב (כמו מעבדי תמלילים) ולפיתוח מסגרת הוראה וחומרי למידה בנושאים הבאים: עיצוב ופיתוח תכנה ובניית חומרי למידה לקורס מבוא לתכנות עצמים.

מהי הגישה המינימליסטית?

מינימליזם היא תיאוריה שצמחה בתחילת שנות ה-1980 כמודל לתיעוד תוכנות ופיתוח מדריכים למשתמש (tutorials). באותם השנים פותחו מחשבים ביתיים – מחשבי ה-PC והם נכנסו לשימוש בעסקים קטנים ובבתים פרטיים. יותר ויותר אנשים ללא רקע במחשבים החלו להשתמש במחשב. כמו כן פותחו יישומים לשימוש כללי, כמו מעבדי תמלילים וגיליונות אלקטרוניים שלוו במדריכים למשתמש. אולם, המדריכים למשתמש שפותחו ליישומים אלו לא התאימו לצרכים של המשתמש הפשוט והשימוש ביישומים אלו (ובמחשב עצמו) גרם לתסכול רב בין המשתמשים. חברת IBM הקימה צוות חוקרים ובראשו John M. Carroll, שתפקידו היה לחקור את אופן השימוש ביישומים שונים. צוות החוקרים, ערך מחקר אמפירי שעקב ותעד את הפעולות שבצעו משתמשי מחשב בעבודתם עם יישומים, את המטרות שלהם ואת הרציונל שעליו בסו המשתמשים את הפעולות שלהם. החוקרים תיארו את המאבק של המשתמשים המנסים להפעיל את היישום כמיטב יכולתם ונכשלים פעם אחר פעם במשימתם. מחקרם שפך אור על תהליך למידה של משתמשי מחשב.

המסקנות העיקריות ממחקרם הם:

- משתמשי מחשב מעדיפים "לעבוד מול המחשב" ולבצע משימות תפעוליות והם חסרי סבלנות לקריאת ההוראות (המדריך למשתמש). כאשר למשתמש חסר ידע מסוים הדרוש לביצוע המשימה, הוא מנסה לאלתר תוך שימוש בידע וניסיון קודמים כדי להמשיך הלאה בביצוע המשימה.
- במקרה של תקלה, המשתמשים בצעו פעולות שוב ושוב ולא ניסו ללמוד מניסיונם. כלומר, הם לא ניסו לבצע ניסיונות שיטתיים ולהבין ולהסיק מסקנות על סמך ידע קודם כדי להבין מה קורה ולהתקדם הלאה.
- במקרים רבים חומרי התמיכה שליוו את התוכנות היו לא מתאימים. למרות שהיה תיעוד רב, הוא התמקד בתיאור המערכת, ולכן המשתמשים שהתמקדו בביצוע משימה לא השתמשו בו (הוא לא עזר להם לביצוע המשימה).
- חומרי התמיכה לא כללו מידע שיסייע בזיהוי ואבחון תקלות (שטירון שלומד מערכת לא יכול להימנע מלבצע אותם) ובכלים להתגבר על תקלות אלה.

כדי לענות על הצרכים של המשתמשים תיאר Carroll עקרונות של הגישה לעיצוב מינימליסטי של מדריכי משתמש, שבאמצעותם יכול משתמש ללמוד באופן עצמאי להשתמש בתוכנה. ניתן לסכם שלושה עקרונות מנחים בגישה זו:

- **למידה תוך כדי עשייה** ולא למידה על ידי קריאה (למידה מכוונת פעילויות).
- העסקת לומד בביצוע **משימה אמיתית** מוקדם ככל האפשר מעודד מוטיבציה ללמידה.
- **מתן תמיכה ללומד לזיהוי ואבחון תקלות** נפוצות והתאוששות מהירה.

השלכות של הגישה המינימליסטית על ניתוח תכנה:

בשנות ה-1980, Carroll חקר ואפיין שתי גישות לפיתוח מערכת תכנה. גישה אחת היא הגישה המובנית לפיתוח מערכת (מודל "מפל מים"), לפיה בניית תכנה נעשה בשלבים ברורים של אפיון, ניתוח, תכנון ויישום, כאשר בכל שלב מוגדרים תוצרים מהווים קלטים לשלב העוקב. גישה שנייה הייתה הגישה "הטבעית". באופן מעשי, טען Carroll, מפתחים אינם מתכננים מערכת מתחילתה ועד סופה בשלב הראשון, אלא תוך כדי תהליך פיתוח הם בונים פתרונות ביניים ופתרונות חלקיים, שמשמשים לבניה של האיטרציה הבאה של המערכת. בנוסף טען Carroll, כי יישום של הגישה המובנית גורם לקשיים, משום שהגדרת המערכת כולה נעשה בשלב הראשון, השלב היחיד בו המשתמש

מעורב, ובסיום מקבל המשתמש מוצר מוגמר שקשה להכניס בו שינויים. מסקנות אלו הובילו את Carroll להגדיר מה שקרוי פיתוח איטראטיבי או פיתוח בגישה אב טיפוס. בפיתוח בגישה איטראטיבית, המערכת אינה מוגדרת בשלב הראשון כסופית, אלא מפותחת בתהליך איטראטיבי תוך כדי מפגש עם המשתמש ומתן ביטוי לדרישות ולהערות שלו שיורחבו למערכת מעוצבת שפוגשת את הצרכים של המשתמש.

התפתחות נוספת בגישה לתכנון מערכות תכנה נבעה מכך, שבמשך השנים התוכנות שפותחו הפכו להיות גדולות ומורכבות, והן מכילות אלפי (אם לא יותר) פעולות, ישויות, קשרים ואירועים. Carroll (1988) טען כי הגישה של עיצוב אב-טיפוס אינה מתאימה למערכות כאלה, ולכן הציע להשתמש בתיאוריות פסיכולוגיות על עיצוב בתחום מדעי – psychological design rational. לפי התיאוריה של היפותזה-היסק לוגי (hypothetic-deductive logic), מנתחים תוצאות שהושגו בתצפיות וניסויים ישירים, ובאמצעות היסק מעלים היפותזה אותה מנסים לאמת בצורה לוגית ע"י נתונים שנגזרו מהניסויים.

מינימליזם ותכנות מונחה עצמים

ב-1988 החל Carroll לחקור סביבות מונחות עצמים ושפת התכנות Smalltalk. Carroll התעניין בשאלה: האם ניתן להרחיב את הגישה המינימליסטית, שהתפתחה כמתודולוגיה לפיתוח מדריכים למשתמש (ביישומים כמו מעבד תמלילים), ולהתאימה גם לפיתוח חומרי למידה המתאימים לסביבות תכנות. לאחר ניתוח הקשיים בהוראת תכנות מונחה עצמים, הציע Carroll מספר המלצות ליישום הגישה המינימליסטית בתחום זה:

- עקרון מפתח של הגישה המינימליסטית הוא להעסיק לומדים בביצוע משימות אמיתיות. אולם בתחום מורכב כמו תכנות מונחה עצמים, משימות אמיתיות מטבען הן גדולות ומורכבות מאוד, ולכן קשה להציגה בצורה יעילה ללומדים טירונים, מה שיכול לגרום לתסכול רב. כדי להתמודד עם בעיה זו פתחו Carroll וצוותו סט של פרויקטים מתחומים שקרובים לתלמידים כמו משחק בלק'ג'ק (Blackjack) ויישמו אותו בסביבת Smalltalk. משימות אלו אפשרו לתלמיד לבצע תהליך פתוח אותנטי הכולל ניתוח, תכנון ויישום.
- מינימליזם מניח שאנשים שעוסקים בביצוע משימה משתמשים באימפרוביזציה וידע קודם. בתחום מורכב כמו תכנות מונחה עצמים, כמות המושגים הראשונית שיש לרכוש היא גדולה מאוד ועד שלומד בדרך כלל גורם לכך פיתוח ועיצוב מערכת מונחת עצמים נדחה לשלבים מאוחרים של הקורס. כמו כן סביבת תכנות כוללת בדרך כלל ספריה גדולה של מחלקות, אותה התלמיד צריך להכיר ולהשתמש. כדי להתמודד עם בעיה זו Carroll וצוותו ממליצים לספק חומרים התומכים בכישורים כלליים הקשורים לתחום הדעת. לדוגמה, כדי לעודד רפלקציה ובקורת עצמית, יש לספק פעילות המאפשרת ניתוח של קוד מקור.
- מינימליזם מדגיש התנסות עצמית של הלומד מהר ככל האפשר. כדי ליישם גישה בתחום דעת מורכב, אמצו Carroll וצוותו את התפישה של Bruner – שימוש בפיגומים, הכולל תכנון של משימות בלתי תלויות זו בזו. כך משימות מורכבות פורקו לתת משימות קטנות יותר שהן בלתי תלויות זו בזו ויש בהן שימוש בידע קודם של התלמיד.
- לסיום כדי לתמוך באבחון והתאוששות מתקלות, התחום דעת מורכב יעיל יותר לספר רמה גבוהה של תמיכה. לדוגמה תיאור של אסטרטגיות לאבחון ופתרון תקלות במקום תיאור של תקלות ספציפיות.

Object-Oriented Design from User Scenarios Rosson, M.B. and Carroll, J.M. (1996).

במאמר זה מתארים Rosson ו- Carroll עקרונות לפיתוח של קורס לעיצוב מונחה עצמים OOD בגישה המינימליסטית, בו התלמידים מועסקים מהתחלה בביצוע משימות אמיתיות, תוך כדי שימוש בתרחישים של משימות משתמש. ניתוח תרחישי משתמש אלו מסייעים ללומד לבנות מודל של עצם, ובהמשך מודלים אלו מוכללים ליצירת מחלקה ובסוף לעיצוב השלם של התכנה המורכבת מאירועים ותרחישי משתמש. המשימות תוכננו כך, שהן לא דורשות ידע קודם רב כדי להבין את המושגים ב-OO ולביצוען השתמשו התלמידים בספריה של מחלקות שהוספה לסביבת העבודה. המשימות הללו משמשות כמנוף ומסייעות לתלמיד להכיר שיטות של תכנון ועיצוב תוך כדי שימוש וניתוח תרחישים המבוססים על אינטראקציות עם המשתמש. במהלך הקורס, התלמידים לומדים לבנות מודלים קונקרטיים של תרחישים, ולנתחם בקונטקסט של המשימה וכך לגלות את היעילות של הפשטה ועקרונות OO. סביבת הפיתוח שבה השתמשו החוקרים כוללת מחולל תרחישים (Scenario Browser with Smalltalk).

תיאור הפעילויות ומבנה הקורס:

- הנחיות ראשוניות: הקורס מתחיל בתיאור קצר ומונחה של הנושאים ב-OOD ובשיטת ההוראה והחומרים שעומדים לרשותם, ולאחר מכן מוצגים מספר נושאים בסיסיים בעיצוב. בכל נושא השתמשו התלמידים בחומרי עזר הכוללים תיאור אופן הביצוע פעילות עיצוב ספציפית והתנסות עצמית על פעילות דומה.
- תכנון תרחישי משימה: תרחישים אלו מתארים משתמש שעסוק בביצוע פעילות משמעותית עם מערכת אמיתית או דמיונית. כייצוג לעיצוב, התרחישים משקפים מושגים וערכים שהם נפוצים בין הרבה אנשי מקצוע ב-Human Computer Interface. התלמידים עוסקים בניתוח התרחישים שהם מוכרים להם כמשתמשים וכך מעוגנת ההוראה בביצוע פעילויות מוכרות וקונקרטיות. דוגמאות בהן השתמשו בקורס כדי להציג תרחישים הם: חנות למוצרי רכב, משחק בלק'ק, איתור ספרים בספריה וכדומה. בשלב זה הניתוח מתמקד בהבנה וניתוח של ממשק משתמש ברמה בסיסית, כך שתלמיד יכול להבין את הפעילות שצריכה להתבצע והן את הרציונל של עיצוב של הממשק.
- זיהוי אובייקטים עיקריים: ניתוח מספר תרחישים משמש לבניית מודל של אובייקט. כדי לסייע לתלמיד להפעיל היוריסטיקה לזיהוי, הוצגה אסטרטגיה לפיה ניתן ל-"סמן את שמות עצם".
- קביעת תחום האחריות של אובייקט (Assigning Object Responsibility): המשימה שניתנה לתלמידים היא "להאניש" (anthropomorphize) את האובייקט שבתרחיש. במילים אחרות, התלמידים משחקים את תפקיד האובייקט הנתון ותוך כדי כך מגדירים את האחריות והתפקידים שלו. פעולת האנשה של האובייקט משמשת כמקור מידע על הידע שאובייקט צריך להכיל (מצב פנימי) על הקשרים שלו עם אובייקטים אחרים ועל התנהגויות שהוא צריך לתמוך.
- בניה תסריט לתרחיש: תרומה תרחיש ל-OOD היא באספקת תוכן והקשר שאתו יכול המתכנן לחקור ולנתח. זה מספק את המבט הפרוצדורלי. במשימות אלו נדרשו התלמידים לבנות תסריט פעולה ("אלגוריתם") לשיטת אובייקט.
- ניצול של ידע משותף למספר אובייקטים: ניתוח האובייקטים שנבנו על סמך תרחישים מאפשר לתלמיד למצוא את המשותף בין מספר מופעים. על האובייקטים שהגדירו התלמידים בשלבים הקודמים, הם בפעילות זו מתבקשים לזהות את המשותף (מבנה והתנהגות) ואת השונה (ערכים), ובהתאם ליצור מחלקה שמכילה את המידע המשותף. תהליך דומה של הכללה של מחלקות יוצר את המושג היררכיה וירוסה.
- פעילות בה תלמידים מנתחים את תפקידו של אובייקט, מוביל לזיהוי סוג האובייקט. לדוגמה סוג אובייקט יכול להיות שתפקידו: לתאם, לבקר, לבנות וכדומה. זיהוי התפקיד של אובייקט מקל על התכנון של מבנה והפעולות שלו.
- פיתוח אובייקט מופשט – התרחישים מנתחים בשני תנאים: פירוק מחדש של אובייקטים שיש בהם חוסר עקביות וסילוק של אובייקטים שקיבלו הרבה תפקידים מכדי שיהיה ניתן לכמס אותם באובייקט מופשט אחד.
- תמיכה בשימוש חוזר ושימוש חוזר על ידי הורשה: שימוש במשימות הכוללות מישק למשתמש המאפשר שימוש באובייקטים מובנים בסביבה.
- לסיום בנית מערכת המבוססת על שימוש במחלקות מוכנות ואובייקטים ומחלקות חדשות שהתלמידים יצרו.

Problem based learning: An instructional model and its constructivist framework. Savery, J. R., and Duffy, T. M. (1995).

למידה מבוססת בעיות

למידה מבוססת בעיות (בקיצור PBL) מוגדרת כשיטת לימוד המציבה את הלומד בתפקיד אקטיבי של פותר בעיה מחיי היום-יום, וכך מסייעת ללומד לפתח כישורים של פותר בעיות עצמאי יחד עם רכישת ידע בתחומי דעת שונים. גישה זו פותחה לפני כ- 30 שנה ע"י Barrows (1986) בבית ספר לרפואה (באוניברסיטת McMaster בקנדה), בעקבות קשיים של סטודנטים לרפואה לרכוש את הידע והניסיון הדרוש לאבחון מקרים קליניים. בניגוד לסביבת למידה מסורתית שהייתה נהוגה, לפיה ארגון הקורס הוא סביב נושאים ותכנים בנושא מסוים והמורה במרכז תהליך הלמידה ומשמש כמקור ידע, Barrows תכנן קורס בו הסטודנטים התמודדו עם ניתוח של מקרים קליניים אמיתיים ותוך כדי כך הם למדו את המושגים והעקרונות הרפואיים ורכשו יכולת פתרון בעיות. עקרונות גישה זו נגזרים מהתיאוריה הקונסטרוקטיביסטית השמה דגש על בניית ידע אקטיבית וביקורתית על בסיס ידע קודם. בגישה זו התלמיד הוא במרכז ותהליך הלמידה מאורגן סביב בעיות אמיתיות, ותוך כדי התמודדות עם הבעיות, התלמיד מגדיר, אוסף ורוכש את הידע בתחומי הדעת (אליהם קשורה הבעיה), ובנוסף הוא רוכש את המיומנויות והניסיון בפתרון בעיות. במשך הזמן, גישה זו אומצה על ידי בתי ספר ואוניברסיטאות בתחומים אחרים.

כאמור, PBL היא גישה פדגוגית המאתגרת תלמידים "ללמוד איך ללמוד", תוך כדי שהם מחפשים פתרון לבעיות שהוצבו בפניהם. התמודדות עם בעיות מעודדת חשיבה ביקורתית ואנליטית, ומספקת הזדמנויות לביצוע פעולות כמו איתור ושימוש במקורות ידע, ניתוח ותכנון פתרונות והערכתם. בנוסף, גישה זו מאפשרת התנסות ורכישת מיומנויות של עבודה שיתופית בקבוצה. לסיכום, למידה-מבוססת-בעיות מסייעת לתלמיד להשיג את היעדים הבאים: יכולת למידה של נושאים חדשים; רכישת מיומנויות פתרון בעיות; חשיבה יצירתית וביקורתית; פיתוח גישה הוליסטית לבעיות ומצבים; יכולת עבודה עצמית; ויכולת לעבוד בקבוצות תוך כדי שיתוף פעולה ושיפור יכולת תקשורת.

ניתן לסכם בטבלה את ההבדלים העיקריים בין סביבת למידה בגישה המסורתית לבין סביבת למידה בגישת PBL

מסורתית	PBL
המורה במרכז תהליך הלמידה	התלמיד במרכז תהליך הלמידה
דגש על רכישת הידע	דגש על שימוש בידע לפתרון בעיות
ידע מאורגן בחלקים שלמידת כולם מהווה את השלם	הידע מוצג בשלמותו בהתחלה ואחר כך מפרקים אותו לחלקים
תפקיד מורה – מעביר ידע	תפקיד מורה – מסייע מנחה
ארגון למידה מובנה וקשיח	ארגון למידה גמיש לשינויים
מורה אחראי על למידת התלמיד	התלמיד אחראי על למידתו

טבלה מס. 1. הבדל בין סביבת למידה בגישה מסורתית ו בגישת PBL

2.2 מודל ההוראה

מודל למידה-מבוססת-בעיות מתייחס לאופן ארגון הקורס, תכנון תהליך ההוראה, בחירת בעיות שסביבן תאורגן הלמידה, הכנת חומרי למידה והוראה, הגדרת תפקיד המורה והתלמיד, יישום והערכה של הישגי התלמידים ושל הקורס. לקורסים המפותחים בגישה זו יש מספר מאפיינים בולטים:

- הבעיות שאיתן מתמודדים לקוחות מחיי היום-יום שמאפשרות התמודדות עם תהליך פתרון אותנטי ככל האפשר.
- בין-תחומיות: הנושאים והמושגים שבהם נוגעים בפתרון הבעיה, הם בדרך כלל קשורים למספר תחומי דעת.
- עבודה בקבוצות.
- תפקיד המורה הוא להנחות את הקבוצות ולסייע לתלמיד ולקבוצה בתהליך פתרון הבעיה.
- שימוש בבעיות שהן ill-structured פתרון בעיה דורש בדרך כלל איסוף מידע ולימוד נושאים ומושגים חדשים. התלמיד צריך לזהות את הידע הדרוש לו ובהתאם להשתמש במקורות ידע נוספים כדי לרכוש ידע זה.

פתוח קורס בגישה זו מבוסס על בחירת בעיות שסביבן יעוגנו כל הנושאים וכל הפעילויות הלימודיות. קיימות מספר גישות לפתוח קורס בגישה זו. החל מהגישה בה לאורך כל הקורס, הלמידה של תלמידים היא למידה עצמית ושיתופית תוך כדי התמודדות עם הבעיה שהוצגה בפניהם ושימוש בחומרי עזר ו- tutorials. במסגרת זו, תפקיד המורה הוא להנחות ולסייע בעבודה של התלמידים. מסגרת אחרת משלבת התמודדות עם בעיות והרצאות. במסגרת זו יש שילוב של עבודה עצמית של תלמידים המתמודדים עם הבעיה יחד עם הרצאות וחומרי עזר (המספקים ידע בתחום תוכן, דוגמאות של פתרון בעיות, tutorial). במקרה כזה תפקיד המורה הוא לא רק להנחות אלא גם לספק ידע.

תכנון בעיה מתאימה הוא קריטי והיא צריכה לכלול את המאפיינים הבאים:

- עיסוק בעיה מהעולם האמיתי.
- בעיה פתוחה שפתרונה אינו חד משמעי אלא מאפשר להתייחס למגוון של אפשרויות.
- דורשת למידה עצמית.
- דורשת עבודת צוות.
- התמודדות עם הבעיה מאפשרת להשיג יעדים לימודיים ולפתח כישורים של פתרון בעיות.
- הבעיה מבוססת על ידע והתנסות קודמים אך פתרונה דורש למידה עצמית של חומר חדש.

פיתוח קורס בגישה זו מכיל את השלבים הבאים:

א. זיהוי תוצרי הלמידה בהתייחס למטרות הלמידה: תכנים וידע בתחומי הדעת הנלמדים, כישורים ומיומנויות לפתרון בעיה (כמו איסוף מידע, ניתוח, סינתזה וכדומה), כישורים ומיומנויות חשיבה גבוהות, כישורים חברתיים (כמו כישורי תקשורת, תפקוד בקבוצה וכדומה).

ב. הגדרת שיטת ההערכה של התלמיד מתייחס להגדרת התוצרים שיבדקו (בשלושת התחומים: ידע, מיומנויות פתרון בעיות ומיומנויות חשיבה גבוהות), הגדרת כלי ההערכה והקריטריונים והציונים לכל קריטריון. בנספח מצורפת דוגמה לכלי הערכה.

ג. תכנון הבעיות כולל את הפעולות הבאות:

- הגדרת המטרות הלימודיות הספציפיות שישגו באמצעות הבעיה (ידע בתחום הדעת, מיומנויות פתרון בעיות ומיומנויות חשיבה מסדר גבוה).
- ניסוח הבעיה, כולל פירוט הקשיים צפויים.
- בדיקת הבעיה - האם הבעיה מאפשרת את השגת המטרות הלימודיות הספציפיות והתאמתה לפי הצורך.
- הוספת מקורות מידע אפשריים.
- הכנת חומרי עזר מתאימים.

ד. תכנון מערך ההוראה (מערך השיעורים המוקצה לפתרון הבעיה). במערך שיעורים זה יש לתכנן את המפגשים כך שיאפשרו לתלמידים העוסקים בפתרון הבעיה, הזדמנות לעבוד ביחידים ובקבוצות. כמו כן, יש להקצות זמן לדיונים עם המורה במליאה (כיתה), וזמן להערכה של תוצרי הביניים והתוצר הסופי.

ה. היישום מתחיל מהצגת הבעיה בפני התלמידים וקביעת את "כללי המשחק".

ו. לסיום, לאחר שהתלמידים סיימו לעבוד על הבעיה, יש לבצע הערכה של הבעיה ולבדוק האם היא השיגה את המטרות שנקבעו לה ואם לא כיצד ניתן לשפרה.

Problem-Based Learning for Foundation Computer Science Courses. Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J.H. and Crawford, K., (2000).

מחקרים רבים בהוראת מדעי המחשב מראים כי לשימוש בגישה זו יש יתרונות רבים ולכן מספר סיבות: מדעי המחשב במהותו כרוך בפתרון בעיות אלגוריתמיות וגישה זו מאפשרת התנסות בתהליך פתרון. סיבה נוספת נובעת מהדינאמיות של המקצוע המדגישה את יכולת הלימוד עצמי של נושאים חדשים כמוימנות בסיסית. כמו כן עבודה במקצוע דורשת יכולת לעבודה בקבוצות, וגישה זו מספקת את ההזדמנות לכך. בסעיף זה נתאר מחקר שערכה קבוצת חוקרים מאוניברסיטת סידי באוסטרליה. החוקרים השתמשו בגישת הוראה זו להוראת קורס מבוא בתכנות מונחה עצמים (Kay, J., et al., 2000). במאמר המסכם מחקר של מספר שנים, הם מציגים את הסיבות שהניעו אותם לבחור בגישה זו, מתארים את מבנה הקורס שפתחו ומציגים ממצאים ממחקר המציג את ההישגים של תלמידיהם שלמדו בגישה זו. תחילה החוקרים מפרטים את הקשיים העיקריים שהיו בהוראת המקצוע מבוא למדעי המחשב בגישה הקונבנציונאלית:

- סטודנטים לא רכשו את המיומנויות לפתרון בעיות ולכן התקשו בהתמודדות עם בעיות. למרות שבתחום מדעי המחשב שרוב עיסוקו בפתרון בעיות, הוכח כי הידע הטוב ביותר נרכש תוך כדי התנסות והתמודדות עם בעיות.
- סטודנטים לא אמצו את הפרקטיקה של איש מקצוע במדעי המחשב. לדוגמה, קשה היה לשכנע תלמידים לתעד, בעיקר משום שתכניות בקורס מבוא היו קצרות.
- היו קשיים פדגוגיים שנבעו מכך שהידע של הסטודנטים שהגיעו לקורס היה הטרונגי וסטודנטים בעלי רקע וידע קודם שונה במדעי המחשב שנאלצו ללמוד ביחד.
- בגישה הקונבנציונאלית לא הייתה התייחסות לעבודה בקבוצות.
- קושי אחר נבע ממערך ההוראה הקונבנציונאלי בו יש חלוקה לשיעורים פרונטאליים ושיעורים מעשיים, כתוצאה מכך לא היו לתלמידים מספיק הזדמנויות לעסוק בפתרון בעיות אמיתיות, אלא רק בבעיות קטנות ומלאכותיות שניתן לפתור אותן בשיעור אחד.

החוקרים החליטו לבצע שני שינויים מהותיים בקורס המבוא: א. לשנות את הפרדיגמה התכנותית ולעבור משפה פרוצדוראלית (שפת פסקל) לתכנות מונחה עצמים (OOP). לשנות גם את גישת ההוראה ללמידה מבוססת פרויקטים. לטענת החוקרים, שבדקו גישות הוראה שונות, גישת ה-PBL מתאימה ביותר להוראת מבוא למדעי המחשב, משום שהיא מספקת הזדמנות לתלמידים לעסוק בפתרון בעיות אמיתיות ולרכוש תוך כדי התמודדות עם הבעיות, מיומנויות של פתרון בעיות וידע ב-OOP.

המלצות לתכנון קורס מבוא לתכנות ב-OOP

החוקרים מיפו את סוגי הידע שסטודנט המסיים קורס מבוא צריך לרכוש ובהתאם הם הגדירו את מטרות הלימודיות של הקורס, ובניהן המטרות הבאות:

א. מטרות הקשורות בידע, בסיום הקורס הסטודנט יוכל:

להגדיר מישק של מחלקה פשוטה וליישמו; להשתמש בעילות בספרית מחלקות נתונה; להשתמש באסטרטגיות קידוד ובדיקה הכוללת סגנון מתאים לכתיבה של תנאים מקדימים ותנאי סיום (pre and post), הרצה ובחינה של מחלקה; להסביר ולהצדיק תכנון במונחים של שיטתיות, כלכליות ואת בחירת אסטרטגיה המתאימה לבדיקה של המחלקה; לקרוא ולהעריך של קוד מקור במונחים של מודולאריות, קוד עצמאי, מישק מחלקה וקשרים בין מחלקות.

ב. שימוש בכישורים מטה קוגניטיביים:

ידע בתכנון הכולל יכולת ניסוח פורמאלי של הבעיות שיש לפתור, הגדרה של נושאים חדשים אותם יש ללמוד, הגדרה של אסטרטגיות ללמידת ידע חדש ויכולת בקרה והערכה של הידע שנרכש; הערכה עצמית הכוללת פיתוח אסטרטגיות המאפשרות בחינה של מושגי תכנות חדשים ושימוש בכלים שונים להערכה עצמית; הרפלקציה על תהליך הפיתוח, וקצב התקדמות ותוצרי הביניים; שימוש במדריכים ובחומר מודפס כדי למצוא מידע דרוש על תוכנות;

יכולת לכתוב דוח על תכנון מחלקה ובחינתה המתאר את המטרות של כל בדיקה והסיבה לבחירה בה; יכולת להציג בעל פה בצורה מסודרת דוח על הפתוח והיישום של המערכת שבנה; יכולת לעבוד בשיתוף פעולה, שימוש בכלים כמו תכנות-ע"י-חוזר (programming by contract) ויכולת תקשורת עם שאר חברי הקבוצה כדי להבטיח שכל אחד יודע את תפקידו ותחום אחריותו, נכונות לתרום למאמץ הכללי של הקבוצה ויכולת להעריך את תרומת הזולת.

ב. מבנה הקורס ב-OOP:

הסמסטר חולק לשלושה חלקים:

בעיה ראשונה (4 שבועות). בחלק זה (startup) התלמידים מתוודעים לגישת למידה מבוססת פרויקטים והאחריות שמוטלת עליהם. לשם כך מוצגת בפני התלמידים בעיה אותה הם צריכים לפתור. בארבעת השבועות ההרצאות בכיתה מתייחסות לתהליך הפתרון בעיה ותיאור של כלים מתאימים, הקניית ידע בסיסי במדעי מחשב הדרוש לפתרון, והתייחסות מפורשת למיומנויות של עבודה בקבוצה. התלמידים מתנסים בביצוע מטלות שונות בהתאם לחומרים שפותחו על ידי צוות ההוראה. הפעילות מסתיימת ברפלקציה על הבעיה שהתלמידים פתרו. למרות שהתלמידים עובדים בקבוצה, ההערכה בשלב זה היא אינדיבידואלית ומטרתה לתת משוב לכל תלמיד על טיב עבודתו.

בעיה שנייה (6 שבועות) בחלק זה (החלק המרכזי של הקורס) התלמידים בחרו בעיה אחת מתוך מגוון בעיות שניתנו להם. כדי לשפר את העבודה בקבוצה, המרצים עודדו יצירת קבוצות שבהן יש רקע שונה לחברי הקבוצה. המורים סיפקו שלבי ביצוע ול"ז לביצוע של כל שלב. כמו כן בנוסף לעבודה הקבוצתית, נתנו הרצאות משלימות בתכנות מונחה עצמים ובהנדסת תכנה, וכל תלמיד נדרש להגיש חלק מהעבודה הקבוצתית וביחד עם עבודה מסכמת של הקבוצה. הערכה בשלב זה כוללת הערכה של ביצועי הפרט וביצועי הקבוצה. בנוסף לבעיה המרכזית, כל תלמיד צריך לבצע גם תרגילים נוספים (קצרים יותר) הקשורים לנושאים שונים.

סיכום והגשת דוחות והדגמות של פתרון בעיה 2 (שבועיים). בחלק זה של הקורס התלמידים מכינים דוחות ומצגות ומדגימים את הבעיות והפתרונות שלהם. חלק זה כולל גם הערכת עמיתים. הדוח שהם מגישים כולל לא רק את התוצר הסופי אלא את שלבי התכנון והעבודה.

ג. מקורות למידה

כדי לתמוך בתהליך הלמידה פותחו מגוון רב של חומרים ללמידה של נושאים ב-OOP ולפתרון בעיות. בנוסף החוקרים פתחו מאגר דוגמאות המתאר תהליך פתרון של בעיות מאותו סוג. הצגת הדוגמאות מאפשרת לתלמידים ללמוד מדוגמאות ולראות דרכים ואסטרטגיות מתאימות להתמודדות עם בעיות. לדוגמה בחלק הראשון הבעיה הייתה לבנות סימולציה המסייעת למנהל בנק, ולכן נתנו לתלמידים דוגמה לבניית סימולציה של מערכת אקולוגית ודוגמה אחרת לבניית סימולציה של מעליות. חומרי עזר נוספים אותם פתחו החוקרים, הם דפים להערכה עצמית ודוגמאות לשימוש בהם.

ד. הכשרת צוות ההוראה

דגש רב הושם על הכשרה של צוות ההוראה כולל סמינרים ופגישות שבועיות, חומרי קריאה וכדומה.

ה. הערכה

תכנון שיטת הערכה כלל:

- קריטריונים המעריכים את הידע בתכנות מונחה עצמים.
- קריטריונים להערכה של תהליך הפתרון ושל התוצר הסופי.
- קריטריונים המעריכים את הכישורים בפתרון בעיות.
- קביעת משקל לפעולות השונות, ובהן מתן משקל שווה לצינונים של מבחנים ועבודה מעשית, ומשקל שווה לצינון של הפרט וצינון של הקבוצה.

דוגמה לבעיה שתוארה בקורס OOP:

מנהל בית התכנה הציג בפני הצוות אליו אתה משתייך פרויקט חדש – בניית סימולציה שתסייע למנהל בנק לבחון שתי שיטות לחישוב עמלות בגין פעולות של לקוחות בחשבון עובר ושב. גביית עמלות יכולה להיות מחושבת בדרכים שונות. שיטה אחת היא לגבות עמלה אחידה על כל פעולה שמבצע לקוח. אפשרות אחרת היא לגבות עמלות רק לסוג מסוים של פעולות מסוימות ומפעולות מסוג אחר לא לגבות כלל עמלה. ניתן לקבוע מדיניות לפיה יש מספר סוגים של לקוחות ולכל סוג לקוח להגדיר גובה עמלה. כדי לאפשר למנהלי הבנק לקבל החלטה לגבי השיטה היעילה לגביית עמלות, יש לבנות סימולציה שתציג ניתוח סטטיסטי של שתי שיטות שונות לחישוב עמלות והשפעתם על חשבונות של הלקוחות.

מנהל הבנק הגדיר מספר משימות שהסימולציה תכלול:

- עדכון פרטי לקוח הכולל מדיניות של גביית העמלה.
- הצגת פירוט הפעולות שבוצעו והעמלות שנגבו לכל פעולה.
- סיכום העמלות לפי סוגים שונים של פעולות וסוגים של לקוחות
- להוסיף מחלקה של טלר הכוללת מישק שיאפשר לבצע את הפעולות הבאות: לקדם כל משתמש בברכה, לשאול לסוג הפעולה המבוקשת: הפקדה/משיכה, לבצע את הפעולה ולדווח ללקוח בסיום על אחוז העמלה
- להוסיף מחלקת לקוח המכילה את פרטי הלקוח הדרושים לבנק.
- לשדרג את המחלקה של הטלר כך שבפעולת משיכה, יפורטו סוגי השטרות שנתנו ללקוח.
- להבטיח שלבנק יוכלו להיות הרבה לקוחות.
- להבטיח שללקוח יכולים הרבה חשבונות.
- להבטיח שללקוח יכול להעביר כסף בין החשבונות השונים שלו.
- להוציא דוחות חודשיים ללקוח.
- להוציא דוחות מסכמים להנהלת הבנק.

הנחיות ביצוע: המורה מיצג את תפקיד המנהל שלך. יש לעבוד בקבוצות של 3-4, כאשר כל תלמיד בקבוצה אחראי על חלק אחד של הסימולציה. משך הביצוע: 6 שבועות. כדי לעזור לך להתחיל, המנהל כבר שילם למתכנת בכיר שכתב מספר מחלקות (class) של בנק בהן תוכל להשתמש. במחלקות הן: חשבון לקוח, פעולה. הבעיה שניתנה לתלמידים כללה מפרט מדויק של מה שתלמיד צריך להגיש, ל"ז ותיאור של שיטת הערכה.

הכותבים ערכו מחקר במשך שלוש שנים בו הם שיפרו את הקורס ואת חומרי למידה. הם בדקו את ההישגים של התלמידים והעמדות שלהם כלפי הקורס. החוקרים מדווחים כי ממוצע הציונים של התלמידים עלה ל- 91 לעומת 63 (שלוש שנים לפני כן) בשאלות תכנות בסיסיות. בנוסף, לטענתם תלמידים שיפרו את היכולת שלהם בהבנת בעיות מורכבות יותר, כמו סיבוכיות, שקודם לכן נחשבו נושאים קשים מאוד ללמידה. שיפור נוסף חל בגישתם של התלמידים, במידת העניין בקורס ובהרגשה כי הם שיפרו את ידיעותיהם במהלך הקורס.

רשימת מקורות

Ellis, A., Carswell, L., Bernat, A., Deveaux, D., Frison, P., Meisalo, V., Meyer, J., Nulden, U., Rugelj, J. and Tarhio, J. (1998). Resources, tools, and techniques for problem based learning in computing. Working Group reports of the 3rd annual SIGCSE/SIGCUE ITICSE conference on Integrating technology into computer science education, Dublin, Ireland , p. 41 - 56

Barrows H.S. & Tamblyn R.M. (1980) Problem-Based Learning: An Approach to Medical Education. New York: Springer Publishing Company.

Ben-Ari. M., (2001). Constructivism in computer science education. Journal of Computers in Mathematics & Science Teaching, 20(1):45—73.

Ben-David Kolikant, Y. (2004). Learning Concurrency as an Entry Point to the Community of CS Practitioners. Journal of Computers in Mathematics and Science Teaching, 23, 1, 21-46.

Carrol, J.M. (1998). Reconstructing Minimalist. from his book Minimalism: Beyond the Nurnberg *Funnel* and reprinted by permission of the publisher (MIT Press).

Carroll, J.M. and Rosson, M.B.(1987) Paradox of the active user, In Interfacing thought: cognitive aspects for human-computer interaction, J.M. Carrol (ed). The MIT press, pp 80-117

Finkle, S. L., & Torp, L. L. (1995). Introductory Documents. Aurora, IL: Illinois Math and Science Academy

Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J.H. and Crawford, K., (2000). Problem-Based Learning for Foundation Computer Science Courses. Computer Science Education.

Rosson, M.B. and Carroll, J.M. (1996). Object-Oriented Design from User Scenarios. Conference on Human Factors in Computing Systems, ACM Press New York, NY, USA

Savery, J. R., and Duffy, T. M. (1995). Problem based learning: An instructional model and its constructivist framework. Educational Technology, v 35, Sep-Oct. pp. 31-38.

ברוקס ז'. ג. , וברוקס מ. ג' (1997). לקראת הוראה קונסטרוקטיביסטית : בחיפוש אחר הבנה. ירושלים: ת"ל ומכון ברנקו וייס.

אתרים העוסקים בלמידה מבוססת פרויקטים:

<http://www.samford.edu/pbl/definitions.html>

<http://www.cet.edu/earthinfo/classroom/teachers/FTtopic1.html>

<http://www.pbli.org/core.htm>

<http://www.pbli.org/faculty/hsb.htm>

אתרים העוסקים בקרל הרבובט:

<http://csis.pace.edu/~bergin/karel.html>

<http://wol.pace.edu/~bergin/KarelJava/Karel++JavaEdition.html>

<http://csis.pace.edu/~bergin/>