

## המחלקה להנדסת מערכות תקשורת

### אוני' בן גוריון בנגב

#### הנחיות הגשה:

1. יש להגיש את קבצי העבודה (12 קבצים סה"כ ללא assignment1.cpp) במערכת ה VPL ולוודא שאין שגיאות קומפילציה במערכת. **אין להגיש את הקובץ assignment1.cpp , קובץ זה נמצא במעטפת ההרצה של VPL.**
2. ההגשה מתבצעת בזוגות, ואין צורך להגיש פעמיים עבור שני בני הזוג.
3. יש להתאים את הפלטים של התוכנית שלכם לפלטים של קובץ ההרצה לדוגמא.
4. הקפידו על הוראות ברורות וקריאות למשתמש לפני כל פעולת קלט.
5. יש להקפיד של שמות המחלקות והשיטות כפי שמצוין בעבודה, הקפידו על אותיות גדולות וקטנות (אי התאמה תגרום לשגיאת קומפילציה בבדיקה האוטומטית).
6. הקפידו על תכנות נכון, זהו חלק מן הציון:
  - a. שמות משתנים בעלי משמעות.
  - b. שימוש חוזר בקוד.
  - c. הקפדה על הזחות.
  - d. שימוש נרחב בהערות באנגלית בלבד. הקפידו לתעד כל שיטה וכל קטע קוד משמעותי.
7. בראש כל קובץ יש להוסיף בהערה את הטקסט הבא:

```
*/Assignment C++: 1
Author: Israel Israeli, ID: 01234567
/*
```

8. עליכם להחליף את הפרטים בפרטים שלכם.
9. כל שאלה יש להפנות לפורום המתאים לשאלות במודל.
10. הארכות יינתנו רק במקרי מילואים, אבל ומחלה חריפה ובצירוף אישורים מתאימים.

# תרגיל מס' 1

## נושא התרגיל: מחלקה – פונקציה בונה, הורסת, public, private

בתרגיל זה אנו נבנה את המחלקות הבאות :

**הערה חשובה:** רשימת המתודות היא **חלקית** בלבד. היא מהווה את המינימום הנדרש ממכם. ייתכן ואף סביר להניח שתצטרכו להוסיף מתודות פרטיות **נוספות** עבור המחלקות המתוארות. חישבו היטב מה באמת צריך בכל מחלקה.

לכל המחלקות חובה להוסיף Constructor ו- Destructor דיפולטיביים (אפילו אם הוא ריק). על הבנאי הדיפולטיבי לאתחל את האובייקט בערכים הגיוניים. כמו כן עליכם להוסיף מתודות set ו-get **לפי הצורך**.

### 1. מכונת ממכר

מכונת ממכר או אוטומט מכירות היא מכונה אוטומטית למכירת מזון (בדרך כלל מזון מהיר או חטיפים) משקאות או חפצים אחרים, שפותחה לראשונה במאה העשרים באנגליה. אתם נדרשים לכתוב תוכנית עבור מכונת משקאות. התוכנית צריכה לעשות את הדברים הבאים:

1. להציג למשתמש את המוצרים השונים שניתן לקנות.
2. לתת למשתמש לבחור את המוצר המבוקש.
3. להציג למשתמש את המחיר של המוצר הנבחר.
4. לקבל כסף מהמשתמש.
5. שחרור המוצר.

המכונה תמכור משקאות קלים: מים, קולה ענבים.  
למכונה יש שני רכיבים עיקריים:  
קופה ורכיב לניהול המלאי במכונה.  
הגדר את המחלקות הבאות:

#### CashRegister

##### Private members:

cashOnHands – integer

##### Public members:

CashRegister() – initiate the cash in the register to 0.

CashRegister(int) - initiate the cash in the register to a specific amount.

getCurrentBalance() - returns value of cashOnHand.

acceptAmount(int) - receive the amount deposited by the customer and update the amount in the register (update cashOnHands).

#### DispenserType

##### Private members:

numberOfItems – Integer

cost – Integer

##### Public members:

DispenserType (int numberOfItems, int cost).

getNoOfItems() - Return the value of numberOfItems

getCost() - Return the value of cost.

makeSale() - Reduce the number of items by 1.

### VendingMachine

#### Private members:

waterDis – DispenserType  
colaDis – DispenserType  
grapesDis - DispenserType  
counter - CashRegister

#### Public members:

VendingMachine () – Initialize the dispensers with the next values:

water(10,5), cola(5,7), grapes(4,6) .

sellProduct(char) – Receive the first character of the wanted product (e.g. 'w' for water).

Ask from the client to pay for the product unless the product is sold out.

Take care of different cases of payment, such as if the client didn't pay enough (see executable).

הערה: המתודה sellProduct מאפשרת למשתמש להזין פעם אחת נוספת (סה"כ פעמיים) את הכסף הנדרש במידה והמשתמש לא הזין מספיק כסף. אם המשתמש לא הזין מספיק כסף גם בפעם השנייה אז לא מתבצעת עסקה ומודפסת הודעה מתאימה. אם המשתמש הכניס מספיק כסף אז מתבצעת עסקה והכסף נכנס לקופה. המכונה לא מחזירה עודף ולכן אם המשתמש הכניס יותר כסף ממה שצריך אז כל הסכום יכנס לקופה.

## 2. טיפוס נתונים מופשט – Abstract Data Type (ADT)

טיפוס נתונים מופשט (Abstract Data Type או ADT) הוא מודל מתמטי עבור קבוצה מסוימת של מבני נתונים בעלי התנהגות דומה. טיפוס נתונים מופשט מוגדר על ידי הפעולות שניתן לבצע עליו ועל ידי מגבלות שחלות על תוצאות הפעולות האלה (או העלות שלהן מבחינת סיבוכיות זמן ומקום).

לדוגמה, מחסנית היא טיפוס נתונים מופשט הכולל פעולת (Push) שמוסיפה איבר, פעולת (Pop) שמסירה את האיבר האחרון שנוסף למחסנית, ופעולת Peek המאפשרת גישה למידע בראש המחסנית ללא הסרתו. בהקשר של ניתוח סיבוכיות של אלגוריתמים העושים שימוש במחסנית, ניתן להוסיף דרישות המקילות את ניתוח הסיבוכיות: זמן קבוע לכל פעולה, ללא תלות בכמות הנתונים במחסנית, ושטח זיכרון קבוע עבור כל איבר.

ניתן לקרוא עוד בקישור הבא: [https://en.wikipedia.org/wiki/Abstract\\_data\\_type](https://en.wikipedia.org/wiki/Abstract_data_type).

עליכם לממש את מחסנית דינמית (מחסנית שהגודל שלה משתנה בהתאם לצורך), לשם כך ממשו את המחלקה הבאה:

### StackNode

#### Private members:

data – int  
next – StackNode\*

### Stack

#### Private members:

top – StackNode\*

#### Public members:

Stack() – Initialize top to NULL.

push(int) - Inserts a new element at the top of the stack.

pop() - Removes the top element on the stack. If stack is empty, print "Stack is empty".

isEmpty() - Returns true if the stack is empty; otherwise, it returns false

peek() - Returns the top element present in the stack without modifying the stack.

In case where the stack is empty, return INT\_MIN (need to include <climits>).

The return type is integer.

### 3. מחלקת תפריט – Menu

מחלקה זו תנהל את המערכת. על מחלקה זו לבצע פעולות קלט/פלט מהמשתמש למעט שיטות שהוגדרו לכך במפורש במחלקות אחרות. יש להציג את התפריט בלולאה עד שהמשתמש יבחר לסיים אותה.

שיטות:

**mainMenu** – שיטה זו תדפיס את התפריט הבא:

המשתמש מקיש	שם הפעולה	הערות
1	כניסה לתפריט מכונת שתייה	לאחר הקשה זו המערכת תפעיל מתודה של תפריט מכונת שתייה
2	כניסה לתפריט מחסנית	לאחר הקשה זו המערכת תפעיל מתודה של תפריט מחסנית
3	יציאה מהתוכנית	לאחר הקשה זו המערכת תדפיס Goodbye! ותצא

**shopMenu** – בעת הפעלת שיטה זו יוצג למשתמש תפריט מכונת השתייה.

המשתמש מקיש	שם הפעולה	הערות
1	מים	הקשה זו תקרא למתודת מכירת מוצר
2	קולה	הקשה זו תקרא למתודת מכירת מוצר
3	ענבים	הקשה זו תקרא למתודת מכירת מוצר
4	יציאה	יציאה מתפריט זה וחזרה לתפריט הראשי

**stackMenu** –

המשתמש מקיש	שם הפעולה	הערות
1	הוספת איבר למחסנית	הקשה זו תוסיף איבר למחסנית בהתאם לקלט שמתקבל מהמשתמש
2	הוצאת איבר מהמחסנית	הקשה זו תשלוף איבר מהמחסנית
3	הצגת האיבר הראשון	הקשה זו תציג את האיבר הראשון במחסנית
4	בדיקת אם המחסנית ריקה	הקשה זו תדפיס הודעה מתאימה בהתאם לתכולת המחסנית
5	יציאה מהתוכנית	יציאה מתפריט זה וחזרה לתפריט הראשי

**ה main** –

מייצרת אובייקט מסוג תפריט ומריצה את המתודה "תפריט ראשי" וזהו.

## הערות:

1. הקלטים יהיו מהטיפוסים החוקיים. ז"א בכל מקום נכניס את הטיפוס המתאים. אנחנו לא מתחייבים שהוא יהיה בטווח מסוים – אלא אם כן נאמר אחרת בשאלה עצמה.
2. אחרי כל הדפסה יש לבצע ירידת שורה.

יש להקפיד על תכנות נכון:

- a. כל הערכים שהם קבועים (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: `define`, `const`, או `enum` בהתאם לצורך.
  - b. יש לרשום הערות בשפע, ובאנגלית בלבד (לכל מחלקה מה התפקיד שלה, התוכנית מה היא מבצעת, כל פעולה לא טריוויאלית להסביר, וכל 2~3 שורות קוד – הערה, כל מתודה מה היא עושה).
  - c. יש להקפיד על כימוס נכון – כל השדות ומתודות השירות ב-`private`, הממשק ב-`public`, חלוקה לקבצים.
  - d. יש להקפיד על הזחות, כיתוב נכון וקריא, ושמות משמעותיים.
  - e. יש לנסות ולייעל את הקוד והתוכנית ככל שניתן. הקפידו על `reuse` בקוד. נקודות ירדו על קוד כפול!
  - f. לפני בקשת קלט, יש להדפיס למשתמש הוראה (איזה קלט מבוקש)
  - g. להזכירם: הנכם נדרשים לתכנת בשפת `C++` אי-לכך, כל שימוש בפונקציות וספריות של שפת `C` אסורה.
3. עבור הצלחת הבדיקות עליכם לוודא שהדפסות זהות לקובץ ההרצה שקיבלתם. (לא יורדו נקודות על שוני ברווחים ושורות).
  4. בהצלחה!