מבוא למדעי מחשב מתקדם
עבודת הגשה מספר 4 – הורשה, פולימורפיזם ו- RTTI

המועד האחרון להגשת העבודה הוא ביום 17.5.15 עד שעה 23:50

הוראות הגשה:

1. **כל שאלה ופניה בנוגע לתרגיל יש להפנות אך ורק למרצה האחראית על התרגיל – חבצלת כהן באימייל: work1sce@gmail.com פניות בכל בדרך אחרת – לא יענו!**
2. הגשה ביחידים בלבד. הגשה בקבוצות גדולות יותר תוביל לציון 0 בעבודה.
3. הגשה דרך מערכת מודול בלבד. שום עבודה לא מתקבלת במייל!
4. **אין לשתף או להעתיק את העבודה או חלקים ממנה.** עבירה על הוראה זו תוביל לציון 0 בעבודה.
5. **כל יום איחור** בהגשת העבודה יוביל ל**הורדה של 5 נקודות** מציון העבודה .לאחר 4 ימי איחור לא תתאפשר כלל הגשת העבודה.
6. הארכות יינתנו אך ורק במקרים חריגים (מילואים או חלילה, אבל על קרובים ומחלה חריפה!) ובצרוף אישורים מתאימים! כמו כן במקרה של ידיעה מוקדמת חובה ליצור קשר עם המרצה האחראית על התרגיל לפחות יומיים לפני חלוף הדד-ליין!

7. בתחילת **כל קובץ** יש להוסיף את התיעוד הבא:

```
/* Assignment: 1
Campus: Beer-Sheva / Ashdod
Author1: Israel Israeli, ID: 012345678
Author2: Geva Givati, ID: 334442221
Date: 17/3/2015 */
```

כמובן שיש להשאיר רק את הקמפוס אליו אתם שייכים , לעדכן את השמות ומספרי תעודות הזהות שלכם , ולעדכן את תאריך ההגשה.

8. את החלק השני יש להגיש מוקלד בקובץ וורד. אין להגיש סריקה של כתב יד!!
9. את הקבצים של החלק הראשון יש לאגד בתיקיה אחת ששמה part1 את הקובץ של החלק השני יש להציב בתיקיה ששמה part2 ואת שתי התיקיות לאגד תחת קובץ **zip** ששמו הוא לפי ת " ז למשל 334442221.zip

חלק 1

במטלה זו כל String שתדרשו לו יהיה מסוג String של מטלה 3.

מעל לכל פונקציה יש לרשום: תפקיד הפונקציה ,פרמטרים שהיא מקבלת , הסבר ליד כל פרמטר מהו תפקידו, ערך מוחזר - מהו הערך המוחזר.

עבור כל פונקציה יש להחליט אם היא וירטואלית או לא ואם היא וירטואלית להחליט אם היא וירטואלית טהורה או לא.

במטלה זו תנהלו חנות שמוכרת מתוקים (SweetItem).

החנות מוכרת מתוקים:
ממתקים (Candy) לפי ק"ג, עוגיות (Cookie) לפי מספר, גלידה(IceCream) לפי כדורים וקוקילידה (Cookielida) (גלידה בין שתי עוגיות).

חשבו היטב איך צריכה להיראות היררכיית ההורשה, האם ההורשה וירטואלית או רגילה, את מי אם בכלל להגדיר כמחלקה אבסטרקטית.

אם מחלקה אמורה לרשת (סוג של...) אסור להגדיר במקום זה הכלה.

המחלקות הן:

- המחלקה SweetItem מכילה שדה מספר פריט שיעקוב אחרי מספר סידורי של הפריטים. שדה זה יתעדכן בכל פעם שיתווסף או ירד פריט.
  - יש לכתוב פונקציה (NoSweets)שתדפיס את מספר המתוקים סך הכל.
- למחלקה Candy יש סוג הממתק (מתוך מבחר מוגדר של ממתקים)משקל בגרמים ומחיר לק"ג .
- למחלקה Cookie יש סוג העוגיה (מתוך מבחר מוגדר של עוגיות) מספר עוגיות ומחיר ליחידה.
- למחלקה IceCream יש טעם (מתוך מבחר מוגדר של טעמים), מספר כדורים ומחיר לכדור.
- מחיר של Cookielida היא עלות של (כדור אחד גלידה ושתי עוגיות)כפול 1.5 .
- לקוקילידה יש גם עמלת ביטול שהיא 10% ממחיר הקוקילידה (כבר טרחנו והכנו לא??). יש להוסיף פונקציה למחלקה קוקילידה שתחזיר את סכום עמלת הביטול (פונקציה זו לא תופיע באף מחלקה אחרת).

המחלקה לקוחות (Customer) :
תכיל: שם לקוח (String) , מספר לקוח שיעקוב אחרי מספר סידורי של לקוחות ומערך מתוקים שייצג את הקניות שלו.

קובץ לקוחות (קובץ Subscription):
קיים קובץ לקוחות שהם חברי מועדון. הקובץ מכיל את שמות הלקוחות (שם ייחודי לכל לקוח).

עבור כל לקוח:

- יש לאפשר הוספת מוצרים (אופרטור =+ שיקבל מוצר) עבור כל מוצר יש להוסיפו רק אם הוא לא קיים (אופרטור =! שיקבל מוצר כדי לוודא שלא מחייבים את הלקוח פעמיים על אותה קניה).
- יש לאפשר גם ביטול מוצר מרשימת המוצרים. אם המוצר הוא קוקילידה יש להוסיף את עמלת הביטול לחיוב הלקוח .
- יש לאפשר הוספת הלקוח כחבר מועדון.

המחלקה SweetShop:

- תכיל את רשימת הלקוחות שקנו בחנות באותו יום (לקוח יכול לקנות בחנות יותר מפעם אחת.).
- תאפשר להוסיף לקוח כחבר מועדון אם הוא לא קיים. עלות חבר מועדון היא 15 ₪.
- בסוף כל קניה של לקוח יש להציג את הסכום לתשלום. הסכום יוצג עם דיוק של ספרה אחת אחרי הנקודה. יש לבדוק האם הלקוח קיים בקובץ אם כן יש לתת לו 5% הנחה ולהציג את הסכום לתשלום לאחר ההנחה עם דיוק של ספרה אחת.

בכל קניה יש להציג תפריט שיציג את האפשרויות עבור כל קניה:
ממתקים, עוגיות וכו' ואפשרות לצאת.
עבור כל בחירה יש לפרט את האפשרויות.

במחלקה **SweetShop** למעט בנא/ים והורס יש רק פונקציה ציבורית אחת:
StartDay("3/4/2015") שתקבל תאריך
3. (ניתן להוסיף מחלקה Date) ותנהל את כל הקניות באותו יום.

בסוף יום העבודה:

- יש להציג עבור החנות כמה לקוחות קנו ומה הפדיון היומי.
- יש להציג את השם, את מספר הלקוח ואת סכום הקניה של הלקוח שסכום הקניה שלו היה הגבוה ביותר. ללקוח זה יש להדפיס כמה כסף הוא הוציא על ומה האחוז של זה מסך הקניה שלו. (למשל הוא קנה ב 100 ₪ וסה"כ הוא הוציא 30 ₪ על עוגיות אז 30% מהקניה שלו זה עוגיות.)
- בעזרת NoSweets יש להדפיס את מספר המתוקים שנמכרו באותו יום.
- יש להוסיף לקובץ (קובץ SumShop) את התאריך ואת הפדיון היומי.

סייבר:

1. שלבו תפריט צבעוני עם צבע ייחודי לכל פריט.

זה הדרייבר:

```
#include "SweetShop.h"
int main()
{
        String day;
        cin>>day;
        SweetShop sweetShop ;
        sweetShop. StartDay (day);
}
```

את היום מכניסים בפורמט day/month/year

חלק 2

.

1. מחלקה אבסטרקטית היא מחלקה שלא ניתן ליצור ממנה אובייקטים. מדוע אנו זקוקים למחלקה כזאת בשפה? לשם מה היא משמשת ומדוע לא ניתן להשתמש במחלקה רגילה (לא אבסטרקטית) במקום?
2. בקוד הנ"ל יש שגיאה. מהו סוג השגיאה? (קומפילציה, קישור, ריצה או לוגית) ומהי השגיאה עצמה? הסבירו!

```
class A{
public:
A(int x) {}
};
class B: public A{
public:
    B() {}
};
int main() {
B b;
}
```

3. אם ההורסים (dtor) של מחלקת הבסיס והמחלקה הנגזרת אינם וירטואליים (שניהם) אזי בעת מחיקת המחלקת הנגזרת – ההורס של מחלקת הבסיס לא יופעל.

נכון / לא נכון
נימוק:

---

4. במה שונה הורשה פרטית מהורשה ציבורית?
5. מהו קישור דינמי (dynamic/ late binding) ומתי משתמשים בו?
6. מדוע בנאי לא יכול להיות וירטואלי?
7. נניח שנתונה מחלקה base שממנה יורשים באופן וירטואלי מחלקות A ו- B. נניח כי נתונה מחלקה AB שיורשת מA ו- מ- B. מי מאתחל את הבנאי של base ומדוע?

8. נתון הקוד הבא:

```
1.  #include <iostream>
2.  using namespace std;

3. class Base {
4.    public:
5.       Base() {cout<<"Base::Base()\n";}
6.      virtual ~Base() {cout<<"Base::~Base()\n";}

7.      virtual void f1() {cout<<"Base::f1()\n";}
8.      void f2() {cout<<"Base::f2()\n"; f3();}
9.      virtual void f3() {cout<<"Base::f3()\n"; f1();}
10. };

11. class Aba: public Base{
12.  public:
13.     Aba() {cout<<"Aba::Aba()\n";}
14.     ~Aba() {cout<<"Aba::~Aba()\n";}

15.     virtual void f1() {cout<<"Aba::f1()\n";}
16.     virtual void f4() {cout<<"Aba::f4()\n"; f3();}
17. };

18. class Ben: public Aba{
19.  public:
20.     Ben() {cout<<"Ben::Ben()\n";}
21.     ~Ben() {cout<<"Ben::~Ben()\n";}

22.     virtual void f1() {Aba::f2(); cout<<"Ben::f1()\n";}
23.     void f3() {cout<<"Ben::f3()\n";}
24. };
```

מלאו את הטבלאות הוירטואליות של המחלקות הנ"ל.

Base class VTABLE	Aba class VTABLE	Ben class VTABLE

<u>שימו לב</u>: חובה לרשום לפני <u>כל</u> מתודה לאיזו מחלקה היא שייכת ע"י הוספת הקידומת ::ClassName לפני כל מתודה.