

סמסטר א' מועד ב'
תאריך: 28/02/2011 14:00
משך הבחינה: 3½ שעות
חומר עזר: כל חומר עזר מותר

מספר זהות:

--	--	--	--	--	--	--	--	--	--

בחינה בקורס: תכנות מכוון עצמים ושפת C++

מרצים: אמיר קירש, ד"ר אלו שגלר

מדבקות

ברקוד

הנחיות כלליות לבחינה:

- המבחן מורכב משני חלקים:
חלק א' כולל 7 שאלות אמריקאיות. משקל כל שאלה 7 נקודות, סה"כ: 49 נק'.
חלק ב' כולל שאלת תכנות שמשקלה הכולל 51 נק'.
חובה לתעד בשאלת התכנות כל פעולה לא ברורה שנעשית.
- בשאלות האמריקאיות יש לסמן תשובה אחת לכל שאלה בטבלה המצורפת. במידה ומספר תשובות נראות נכונות יש לסמן את התשובה הנכונה ביותר.
- בסיום המבחן יש לרשום מס' ת.ז. על גבי טופס המבחן, לוודא שטבלת התשובות האמריקאיות נמצאת יחד עם טופס המבחן, ולהגישם בתוך מחברת הבחינה.
- המבחן הינו עם חומר פתוח. כל חומר מותר למעט מחשבים ניידים למיניהם. אין להעביר חומר עזר בין תלמידים במהלך המבחן.
- נא לכתוב בכתב קריא ולא מחובר.

בהצלחה !

חובה לספק
חסבר עבור
תשובה ו'

טבלת תשובות לחלק האמריקאי

א	ב	ג	ד	ה	ו
<u>שאלה 1</u>					
<u>שאלה 2</u>					
<u>שאלה 3</u>					
<u>שאלה 4</u>					
<u>שאלה 5</u>					
<u>שאלה 6</u>					
<u>שאלה 7</u>					

חסברים לתשובות

חובה לספק חסבר במידה ונבחרה תשובה ו' (אף תשובה אינה נכונה).
מותר לצרף חסבר גם עבור תשובות אחרות. אמנם **רק** תשובה נכונה תזכה בניקוד עבור כל שאלה, אולם ניתן יהיה להסתמך על החסבר במסגרת ערעור, אם יידרש. מומלץ לצרף חסבר לתשובה אמריקאית במיוחד במקרים בהם נראה לך שתשובתך דורשת חסבר או נימוק.

שאלה 1 :

שאלה 2 :

שאלה 3 :

שאלה 4 :

שאלה 5 :

שאלה 6 :

שאלה 7 :

חלק א' – שאלות אמריקאיות (49 נק' – 7 נק' לכל שאלה)

שאלות 1-4 מתייחסות לקטע הקוד הבא:

```
1. int main()
2. {
3.     Person * pPerson;
4.     ifstream file("data.dat", ios::binary);
5.     pPerson = PersonFactory::getPersonFromFile(file);
6.     cout << *pPerson << endl;
7.     Student * pSt = dynamic_cast<Student*>(pPerson);
8.     cout << pSt->getSchoolName() << endl;
9.     file.close();
10.    return 0;
11. }
```

חנח: dynamic_cast שנכשל זורק exception מסוג bad_cast

שאלה 1

אילו מן חדרישות הבאות תכרחית על מנת ששורות 3 עד 6 יעברו קומפילציה?
שים לב: יתכנו דרישות נוספות שאינן מופיעות למטה).

- א. שלמחלקה Person יהיה אופרטור *
- ב. שלמחלקה Person יהיה Empty Constructor
- ג. שהמחלקה Person תדע להדפיס את עצמה באמצעות << ל-ostream
- ד. תשובות ב' ו-ג'.
- ה. תשובות א', ב' ו-ג'.
- ו. אף תשובה אינה נכונה (אף טענה אינה הכרחית או ישנה יותר מטענה הכרחית אחת).

שאלה 2

אילו מהטענות הבאות לגבי התוכנית אינה נכונה?

- א. אם ה-Person הראשון שנקרא מחקובץ איננו מסוג Student, הקובץ לא ייסגר.
- ב. הפונקציה getPersonFromFile במחלקה PersonFactory היא פונקציה סטטית.
- ג. שורה מס' 7 מפעילה Copy Constructor של Student.
- ד. שורה מס' 7 מפעילה אופרטור השמה של Student.
- ה. יכול להיות שהפונקציה getSchoolName() חוגדרה כפונקציה const.
- ו. אף תשובה אינה נכונה (כלומר – הטענות בסעיפים א' עד ח' כולן נכונות או שישנה יותר מטענה לא נכונה אחת).

שאלה 3

להלן חתימות מוצעות עבור הפונקציה `PersonFactory::getPersonFromFile` שנקראת בשורה 5:

```
class PersonFactory {
public:
    ...
    // [i]
    const Person* getPersonFromFile(ifstream& inFile) const;
    // [ii]
    static Person* getPersonFromFile(ifstream& inFile) const;
    // [iii]
    static Person& getPersonFromFile(ifstream& inFile);
    // [iv]
    static Person getPersonFromFile(ifstream& inFile);
    // [v]
    Person* getPersonFromFile(ifstream& inFile);
};
```

אילו מהחתימות יכולה להיות חתימה אפשרית עבור הפונקציה?

- א. ii
- ב. ii, iii
- ג. ii, iii, iv
- ד. i, v
- ה. iii, iv
- ו. אף תשובה אינה נכונה.

שאלה 4

אילו מן הדרישות הבאות הכרחיות על מנת ששורה 8 תעבור קומפילציה?

(שים לב: יתכנו דרישות נוספות שאינן מופיעות למטה).

- א. שהמחלקה `Student` תדע להדפיס את עצמה באמצעות `<<` ל-`ostream`.
- ב. שלמחלקה `Student` יש אופרטור `>`.
- ג. שהמחלקה `Student` הגדירה פונקציה `getSchoolName()`, שאיננה `const`, בקטע ה-`public`.
- ד. שלמחלקה `Student` יש פונקציית `public` שנקראת `getSchoolName()` – או שהגדירה אותה או שקיבלה אותה בחורשה.
- ה. שלמחלקה `Student` יש פונקציית `public` שנקראת `getSchoolName()` – או שהגדירה אותה או שקיבלה אותה בחורשה, ושפונקציה זו איננה `const`.
- ו. אף תשובה אינה נכונה (אף טענה אינה הכרחית או ישנה יותר מטענה הכרחית אחת).

שאלות 5-7 מתייחסות לקטע הקוד הבא:

```

1.  template<class T>
2.  class A
3.  {
4.  public:
5.      static T* max;
6.      A(T* t)
7.      {
8.          if(max == NULL || *t > *max) {
9.              delete max;    // deletion of NULL is OK
10.             max = t;
11.         }
12.         else {
13.             delete t;
14.         }
15.     }
16.     T& operator*() {return *max;}
17.     operator T*() {return max;}
18. };
19.
20. template<class T> T* A<T>::max;
21.
22. int main()
23. {
24.     A<int> a1 = new int(5);
25.     A<int> a2 = new int(4);
26.     cout << *(A<int>::max) << endl;
27.     return 0;
28. }

```

שאלה 5

מורן, לירן, שירן ונברן התכוונו למבחן ועברו על הקוד לעיל.

מורן טענה: התוכנית תדפיס 5.

לירן טען: התוכנית תדפיס את הכתובת של ה-5 שחוקצה בשורה 24.

שירן טענה: בתוכנית יש זליגת זכרון, כלומר לאחר שורה 28 ישנו זכרון שלא שוחרר.

נברן טען: אם היו מוספים את המילח explicit ל-c'tor בשורה 6, הקוד לא היה עובר קומפילציה.

מי מהחברים צודק?

- | | |
|----------------------|-------------------------|
| א. מורן ונברן. | ד. לירן ונברן. |
| ב. לירן ושירן. | ה. לירן, שירן ונברן. |
| ג. מורן, שירן ונברן. | ו. אף תשובה אינה נכונה. |

עבור שאלות 6-7 חוגדרה מחלקה נוספת:

```
1. class StringA: public A<string>
2. {
3. public:
4.     StringA(string* s): A<string>(s) {}
5. };
```

שאלה 6

מה יעשה ה-main הבא:

```
1. int main()
2. {
3.     string str = "Hello";
4.     StringA a(&str);
5.     cout << *(a.max) << endl;
6.     return 0;
7. }
```

- א. ידפיס Hello למסך.
- ב. התוכנית לא תעבור קומפילציה, כיון שהפרמטר שנשלח בשורה 4 אינו הקצאה דינמית.
- ג. התוכנית לא תעבור קומפילציה בגלל בעיה בשורה 5.
- ד. התוכנית תעוף בזמן ריצה, לפני שורה 5, כיון שהפרמטר שנשלח בשורה 4 אינו הקצאה דינמית. לכן גם לא יודפס דבר.
- ה. התוכנית תדפיס Hello למסך, אבל תעוף לפני סיומה בשל ניסיון לשחרר זכרון שלא חוקצה דינמית.
- ו. אף תשובה אינה נכונה.

שאלה 7

מה יעשה ה-main הבא :

```
1.  int main()  
2.  {  
3.      StringA s1 = new string("Hello");  
4.      A<string> s2 = s1;  
5.      A<string> s3 = new string("World");  
6.      cout << *(s1.max) << endl;  
7.      return 0;  
8.  }
```

א. ידפיס Hello למסך.

ב. ידפיס World למסך.

ג. התוכנית לא תעבור קומפילציה, שורה 4 אינה חוקית.

ד. התוכנית תעוף בזמן ריצה, בעקבות שורה 4. לכן גם לא יודפס דבר.

ה. התוכנית תעוף בזמן ריצה, בעקבות שורה 5. לכן גם לא יודפס דבר.

ו. אף תשובה אינה נכונה.

חלק ב' – שאלת תכנות (51 נק')

לפני התחלת כתיבת הפתרון יש לקרוא את סעיף ב במלואו

בית המכירות הפומביות סיבת'וס מארגן מכירות פומביות, מכירות VIP ומכירות מכרז. עליכם לתכנן ולממש מערכת אשר תנהל את המידע על המוכרים, הקונים, הפריטים והמכירות, ואשר תנהל את המכירות על-פי החוקים שיוגדרו לחלן.

שימו לב: המכירות מתבצעות בזירה פיסית (בחדר) ולא במערכת מחשב. מערכת המחשב אמורה לתעד את הפריטים לפי התרחשותם. תינדר נדרש לממש את המנוע הפנימי ללא ממשק משתמש.

נתוני הפריטים שעומדים למכירה כוללים: שם הפריט, סוג הפריט (ציור, פסל, תכשיט, יודאיקה, וכו' – יש לאפשר חוספה של סוגים חדשים), השנה בה הפריט נוצר, שם היוצר, המוכר, סכום התחלתי (סכום מינימום), סכום סגירה (זה ששולם בפועל, 0 כל עוד לא התבצעה מכירה), סוג המכירה אליו מיועד הפריט: מכירה פומבית, מכירה VIP או מכרז.

פרטי קונה כוללים: מס' במערכת, שם, סוגי הפריטים המעניינים את הקונה (למשל: ציורים ופסלים. קונה יכול להיות מעוניין במספר סוגי פריטים אפשריים, באחד, באף אחד או בכלם), וכן – אוסף כל הרכישות שביצע הקונה.

פרטי מוכר כוללים: מס' במערכת, שם, וכן – אוסף כל הפריטים שהוצעו/מוצעים למכירה בעבר ובעתיד, כולל כאלו שטרם נמכרו.

יש לציין כי מוכרים יכולים להיות גם קונים ולחיפך. בית המכירות מעוניין שהמס' במערכת שמייצג קונה יהיה בדיוק אותו המספר שמייצג את המוכר (אם מדובר באותו אדם). בנוסף, מוכר קיים במערכת יכול להשתתף כקונה, באמצעות מספרו במערכת, בלי צורך ברישום חדש, וגם לחיפך – קונה קיים יכול להציע פריט למכירה בלי צורך ברישום חדש במערכת.

מכירה פומבית מתבצעת באופן הבא: המחיר הגבוה ביותר שהוצע ידוע לכל. ניתן להציע הצעה חדשה רק אם היא גבוהה מההצעה האחרונה. בכל מקרה לא ניתן להציע הצעה שנמוכה ממחיר המינימום. בכל רגע נתון מנהל המכירה יכול לסגור את המכירה ואז הזוכה הוא המציע האחרון (שכמובן הציע את ההצעה הגבוהה ביותר).

מכרז מתבצע באופן הבא: המחיר הגבוה ביותר איננו ידוע. כל מציע מגיש את הצעתו. בכל מקרה לא ניתן להציע הצעה שנמוכה ממחיר המינימום. בכל רגע נתון מנהל המכירה יכול לסגור את המכירה ואז הזוכה הוא המציע שהציע את ההצעה הגבוהה ביותר.

מכירת VIP מתבצעת באופן הבא: במכירה יכולים להשתתף אך ורק קונים שקנו עד כה בלפחות 100,000 ש"ח. קונים שעונים על תנאי זה יכולים להצטרף למכירה. ברגע שהמכירה התחילה לא ניתן לצרף קונים נוספים. בתחילת המכירה הכרוז מכריז על מחיר מקסימום שאותו לא ניתן לעבור. המכירה מתבצעת כסבב בין הקונים, לפי סדר הרשמתם (מתחילים בראשון שנרשם, עוברים לשני וכו'). כל קונה יכול לבצע אחת מהפעולות הבאות: (א) להציע מחיר גבוה מההצעה האחרונה שהוצעה (בכל מקרה לא ניתן להציע הצעה שנמוכה ממחיר המינימום); (ב) לדלג על תורו. קונה שדילג על תורו שלוש פעמים יוצא מהמכירה; (ג) להציע את מחיר המקסימום – המכירה מסתיימת מיידית וקונה זה הוא הזוכה; -- בסיום כל סבב יכול עורך המכירה להחליט על סיום המכירה (אבל רק בסיום סבב). כל עוד לא החליט עורך המכירה על סיומה וכל עוד לא הוצע

מחיר המקסימום, המכירה ממשיכה. ניתן להניח שגם אם הוצע מחיר המקסימום נדרשת פעולה יזומה בתוכנה כדי לסגור את המכירה (ובכל מקרה המערכת לא תסכים לקבל הצעות חדשות).

יום מכירות בסיבת'וס כולל: 5 מכירות פומביות (אחת אחר השניה), לאחריו 7 מכרזים (אחד אחר השני) ובסיום 4 מכירות VIP (אחת אחר השניה).

בכל מכירה מכל סוג שהוא, מוצע כמובן פריט אחד בלבד. הפריטים מוצעים לפי סדר הכנסתם למערכת (ראשון – ראשון, שני – שני וכו'), כמובן כל פריט לפי סוג המכירה שנקבע לו. בכל מכירה מכל סוג שהוא, יתכן שאף אחד לא הציע הצעה ואז אין קונה. במצב זה הפריט יעבור לסוף אוסף הפריטים הממתינים למכירה, מסוג מכירה זה (מכירה פומבית / מכרז / VIP). נדרשת פונקציה:

```
void AuctionHouse::createNewSalesDay() throw(SalesDayAlreadyExistsException);
```

הפונקציה הנ"ל איננה פונקציה סטטית.

קריאה לפונקציה תזרוק Exception מסוג SalesDayAlreadyExistsException במידה וישנו יום מכירות במערכת שטרם הסתיים. לאחר הפעלת הפונקציה ייווצר במערכת יום מכירות, אך לא יופעל עדיין.

נדרש כי המערכת תוכל להוציא לפני קיום יום מכירות, רשימה של כל הקונים הרשומים שיכולים להתעניין באיזשהו פריט מהפריטים המוצעים. הרשימה צריכה להיות בצורה הבאה: עבור כל מס' קונה, אוסף מספרי הפריטים שמוצעים במכירה ויכולים לעניין אותו (בהתאם לתחומי העניין שעליהם הצהיר הקונה) – כמובן רק לקונים שיש עבורם לפחות פריט אחד רלבנטי. לצורך הפקת הרשימה שתוארה למעלה, נדרשת פונקציה:

```
map<long, list<long>> AuctionHouse::getRelevantBuyersForSalesDay()const  
throw(NoSalesDayDefinedException);
```

הפונקציה הנ"ל איננה פונקציה סטטית.

קריאה לפונקציה תזרוק Exception מסוג NoSalesDayDefinedException במידה ולא קיים במערכת יום מכירות חדש (כזה שטרם החל).

הפונקציה מחזירה map שממפה מספרי לקוח לרשימת מספרי פריטים רלבנטיים לכל לקוח.

בנוסף תוגדר הפונקציה:

```
void AuctionHouse::startSalesDay() throw(NoSalesDayDefinedException);
```

הפונקציה הנ"ל איננה פונקציה סטטית.

קריאה לפונקציה תזרוק Exception מסוג NoSalesDayDefinedException במידה ולא קיים במערכת יום מכירות חדש (כזה שטרם החל).

הפונקציה תתחיל את אירוע יום המכירות, ע"י התחלתה של המכירה הראשונה. יש לזכור שאין לכתוב למסך או להמתין לקלט מהמשתמש! עליך לממש את המנוע הפנימי בלבד! המשמעות של התחלת מכירה הינה שניתן להתחיל להציע הצעות עבור המכירה ע"י קריאה לפונקציה.

סעיף א' (25 נקודות)

כתוב את הגדרת כל המחלקות הנדרשות לפתרון (כל ה-prototypes, ללא מימושים).
הקפד על שימוש נכון ב-const, private, protected, public.

סעיף ב' (20 נקודות)

- כתוב מימושים של (נא לציין במחברת בצמוד לפתרון של כל תת-סעיף – סעיף ב1, סעיף ב2, וכו') :
1. הקמה של מוכרים/קונים חדשים והכנסתם למערכת (פרמטרים הכרחיים בלבד) - (1 נק').
 2. חוספת תחום עניין לקונה ספציפי (סוג פריט נוסף שמעניין את הקונה) - (1 נק').
 3. הקמה של פריטים למכירה והכנסתם למערכת - (1 נק').
 4. חוספת תחום עניין חדש למערכת (סוג פריט לצורך סיווג תחומי עניין, למשל "כלי נגינה עתיקים") - (1 נק').
 5. יצירת יום מכירות חדש והתחלתו (לפי החתימות המדויקות שתוארו בעמוד הקודם) - (3 נק').
 6. כל הפונקציות הרלבנטיות לקיום המכירות מחסוגים השונים - (8 נק').
 7. פונקציות רלבנטיות לסיום מכירה – ברמה של יום המכירות (הפעלת קריאה לסיום מכירה ברמת יום המכירות תבצע קריאה לסיום המכירה הנוכחית ותעבור למכירה הבאה) וכן ברמה של המכירה עצמה (סגירת המכירה והחלטה על הקונה, עדכון נתונים וכו') - (5 נק').

שים לב:

- כל הפונקציות הינן ללא קלט/פלט!
- נדרש מימוש מדויק ומלא של כל פונקציה.

סעיף ג' (6 נקודות)

כתוב מימוש של הפונקציה:

```
map<long, list<long>> AuctionHouse::getRelevantBuyersForSalesDay()const  
    throw(NoSalesDayDefinedException);
```

שים לב:

- הפונקציה הינה ללא קלט/פלט!
- נדרש מימוש מדויק ומלא של הפונקציה.

סוף !