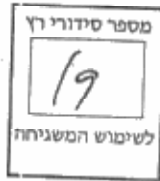


אוניברסיטת חיפה

357089

מחברת



מחברת מס' _____

מתוך _____ מחברות

שם המורה ברוך

בחינה בקורס תכנון אדריכלות

תאריך הבחינה 4.7.2004

החוג אדריכלות

הערכת הבוחן _____

הציון _____

חתימת הבוחן _____

הערות הבוחן _____



**מבחן מועד א' בקורס: "תכנות מונחה עצמים"
סמסטר אביב תשס"ד**

- משך המבחן: שעתיים וחצי.
- מותר חומר עזר כתוב בלבד.
- ליד כל סעיף מצוין מספר הנקודות.
- את התשובות יש לכתוב על גבי טופס הבחינה בלבד. ניתן להעזר במחברת כטייטה.
- הבוחן כולל שלושה חלקים. יש לענות על כל השאלות.
- חלק מהשאלות דורשות בחירת תשובה אחת מתאימה מבין תשובות מוצעות.
- חלק מהשאלות דורשות השלמת מלל, הסבר או הבהרה.
- חלק מהשאלות דורשות השלמת קוד.
- הניקוד לכל שאלה מופיע בצידה.

ניקוד:

30	/ 3	שאלה 1	חלק א (39 נק')
	/ 3	שאלה 2	
	/ 3	שאלה 3	
	/ 3	שאלה 4	
	/ 3	שאלה 5	
	/ 3	שאלה 6	
	/ 3	שאלה 7	
	/ 3	שאלה 8	
	/ 3	שאלה 9	
	/ 3	שאלה 10	
	/ 3	שאלה 11	
	/ 3	שאלה 12	
	/ 3	שאלה 13	
8 / 10	שאלה 14		
10 / 10	שאלה 15		
8 / 10	שאלה 16		
34	4 / 5	שאלה 17.1	חלק ג (36 נק')
	10 / 10	שאלה 17.2	
	3 / 3	שאלה 17.3.1	
	3 / 3	שאלה 17.3.2	
	2 / 3	שאלה 17.3.3	
	3 / 3	שאלה 17.3.4	
	3 / 3	שאלה 17.3.5	
	3 / 3	שאלה 17.3.6	
3 / 3	שאלה 17.3.7		
90	/ 105	סה"כ	



מאיזה בעייתיות סובל קטע הקוד הבא:

```
1. #include <stdio.h>
2. class X { ... //some implementation };
3. void main() {
4.     const X* px = new X("arg", 8);
5.     free((void*)px);
6. }
```

- א. שגיאת הידור (קומפילציה) מאחר ולא ניתן לשחרר זיכרון דינאמי ע"י free (שורה 5) אלא ע"י delete.
- ב. שגיאת הידור (קומפילציה) מאחר ולא ניתן אובייקט רגיל למצביע שהוא const (שורה 4).
- ג. שחרור זיכרון לקוי מאחר ולא יופעל destructor לאובייקט.
- ד. הקצאת זיכרון לקויה מאחר ולא ניתן להקצות אובייקט X עם פרמטרים.
2. נתונות שלושת הפונקציות הגלובליות הבאות המחשבות ציון לסטודנט במבחן ב- oop:

```
(1) int oopTest ( long studentID ) { return (50+rand()%10); }
(2) const char* oopTest ( char* const studentID ) { return "50"; }
(3) long oopTest ( long studentID, long copyingFrom=0 ) { return (30+rand()%10); }
```

- א. הפונקציות מוגדרות היטב ועוברות הידור ללא כל בעייה.
- ב. הקוד לא יעבור הידור בגלל שכפול של הגדרה (multiple definition) של הפונקציות.
- ג. הקוד לא יעבור הידור בגלל הערך המוחזר באחת הפונקציות שאינו תואם את הגדרתה.
- ד. הקוד לא יעבור הידור בגלל שאין שימוש בערכים המועברים לפונקציות.

3. היכן לא כדאי אף פעם להגדיר אובייקט זמני מטיפוס X : X temp; ?

- א. בבנאי כלשהו של המחלקה X.
- ב. במפרק של המחלקה X.
- ג. באופרטור ההשמה (=).
- ד. תשובות (א) ו- (ב) נכונות.

4. במידה וכל הבנאים (constructors) של מחלקה X מוגדרים בחלק הפרטי (private) של המחלקה,

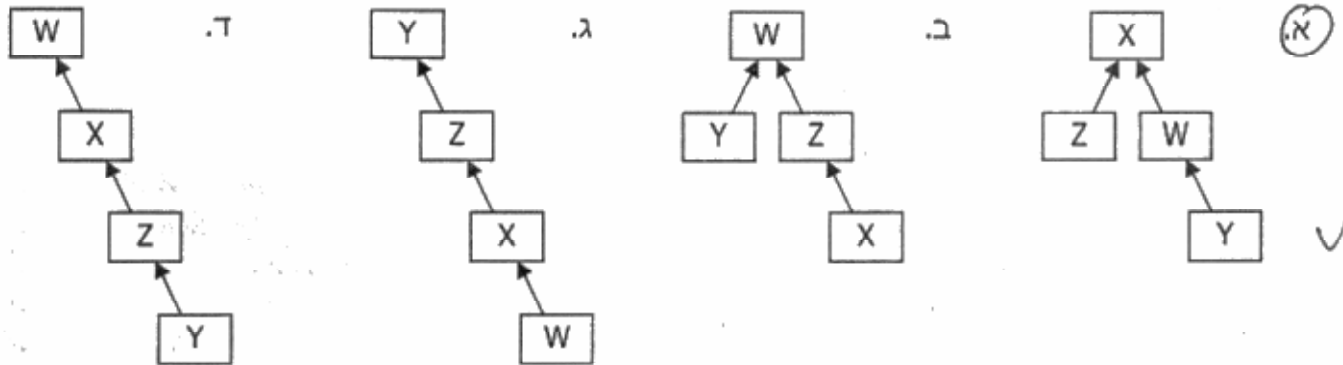
- א. לא ניתן להגדיר אובייקט מטיפוס X בשום צורה שהיא.
- ב. רק מתודה סטטית של המחלקה X יכולה ליצור אובייקט מטיפוס X.
- ג. רק מתודה שאינה סטטית של אובייקט מטיפוס X יכולה ליצור אובייקט נוסף מאותו טיפוס.
- ד. אף תשובה אינה נכונה.

5. איזו מתבניות (templates) הבאות אינה תקינה (לא בהכרח תעבור הידור)?

```
template <class T, class S > T& f(int& t1, S t2) { return t1; }
template <class T, class S=int> class L{ int f(T& t, S& s){ return 0; } };
template <class T=int, class S > class V{ public: V(S&); };
template <class T> class M{ T** t; };
```

נתונות 4 מחלקות X, Y, Z ו-W אשר יש ביניהן יחסי ירושה, ואף ייתכנו יחסי הכלה. כאשר מפורק אובייקט מטיפוס Y נקראים המפרקים הבאים של המחלקות בסדר הבא משמאל לימין:

$\sim X()$, $\sim W()$, $\sim X()$, $\sim Z()$, $\sim Y()$
 מה מהגרפים הבאים יכול לתאר את יחסי הירושה בין המחלקות הנ"ל?



7. אילו אופרטורים צריכים להיות מוגדרים על מחלקה לטיפול במספרים שלמים גדולים בשם BigInt כדי שהקוד הבא יעבור הידור (קומפילציה):

```
BigInt b(1234);
const char* intAsStr = b + 15;
```

- א. operator int -1
- ב. operator+ -1
- ג. operator= -1
- ד. operator int -1

8. להלן מימוש של תבנית שמטרתה לבצע חישוב מכסימום בין שלושה איברים מאותו טיפוס:

```
template <class T>
T MyMax(T left, T mid, T right)
{
    int big = (left>mid) ? left : mid;
    return ((big>right) ? big : right);
}
```

- א. תקבל שגיאת הידור מאחר ולא ניתן לדעת האם הוגדר על הטיפוס המבוקש >operator.
- ב. התבנית תחזיר את הערך 4.7 כאשר נקרא לה באופן הבא: MyMax(5.9, 5.0, 4.7)
- ג. התבנית תחזיר את הערך 5.0 כאשר נקרא לה באופן הבא: MyMax(5.9, 5.0, 4.7)
- ד. התבנית תחזיר את הערך 5.9 כאשר נקרא לה באופן הבא: MyMax(5.9, 5.0, 4.7)

9. איטרטור הינו:

- א. סוג של מצביע לאיבר במכולה (container).
- ב. מחלקה שמוגדרות עליה פעולות ++, --, *, ->.
- ג. מחלקה בעלת ידע וגישה לייצוג הפנימי של המכולה בה היא מוגדרת.
- ד. מחלקה שבעזרתה ניתן לפנות לשדות בתוך המכולה בה היא מוגדרת.



בניח כי הגדרנו רשימה מקושרת של stl באופן הבא:
`list<list<int*>*> mylist;`
 כיצד נגדיר איטרטור העובר על איברי רשימה זו?

- א. `mylist::iterator it;`
- ב. `list<int*>::iterator it;`
- ג. `list<list<int*>*>::iterator it;` ✓
- ד. `list<int*>*>::iterator it;`

11. נתונה הפונקציה הבאה:

```

1. int& f (int& i) {
2.     switch(i){
3.         case(0) : { int x = 6;           return x; }
4.         case(1) : { static int y = 7;    return y; }
5.     }
6.     int* z = new int(8);
7.     return *z;
8. }
```

- א. בשורה (3) יש שגיאת הידור (קומפילציה) מאחר ומוחזר ערך לוקלי. ✓
- ב. בשורה (4) יש שגיאת הידור מאחר ומוחזר ערך סטטי.
- ג. בשורה (7) יש שגיאת הידור מאחר ומוחזר ערך המוקצה דינאמית.
- ד. אין שגיאות הידור עבור הפונקציה הנ"ל, אך תתכן אזהרה (warning) כלשהי. X

12. מה תהיה ההדפסה בסוף קטע הקוד הבא:

```
inline int f (int& x) { x++; return x; }
```

```

void main(){
    int x, y=100, *z=&y;
    x=f(y);
    *z=f(x);
    y=f(*z);
    cout << x << ',' << y << ',' << *z;
}
```

- א. 102,102,102
- ב. 103,103,103
- ג. 102,103,103 ✓
- ד. 102,103,102

13. תהי B מחלקת יסוד כלשהי המגדירה בנאי יחיד המקבל כפרמטר int. תהי D מחלקה הנגזרת ממנה ומגדירה בנאי יחיד המקבל כפרמטר String. כיצד יכול להיות מוגדר הבנאי של מחלקה D?

- א. `D::D(String s) { B(8); }`
- ב. `D::D(String s) : B(8) {}` ✓
- ג. `D::D(String s) : B=8 {}`
- ד. `D::D(String s) {}`



לקב'

בחלק זה 3 שאלות קצרות שוות ניקוד (10 נקודות כל אחת). יש לענות במקום הנדרש בדף בחינה זה.

שאלה 14:

נתונה המחלקה הבאה:

```
#include <list>
using namespace std;

class MyList : public list<int> {
public:
    void join();
};
```

עליכם לממש את המתודה join אשר מקבצת ערכים דומים יחדיו למופע הראשון שלהם ברשימה. לדוגמא:

12→2→5→2→11→3→2→11→7
12→2→5→11→3→7

הרשימה המקורית:
after call to join();

עליכם לממש את המתודה בהמשך להגדרת המחלקה. אין לכתוב פונקציות עזר, אך ניתן לקרוא לפונקציות/מתודות קיימות.

```
void MyList::join()
```

{

```
list<int>::iterator it1, it2;
```

```
for(it1 = begin(); it1 != end(); it1++)
```

{

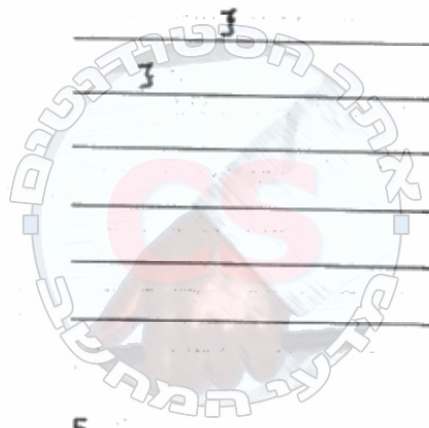
```
for(it2 = it1; it2 != end(); it2++)
```

```
if (*it1 == *it2)
```

```
remove(*it2);
```

}

X



פניכם קטע קוד ובו 3 שגיאות הידור (קומפילציה). מהן השגיאות וכיצד ניתן לתקנן?

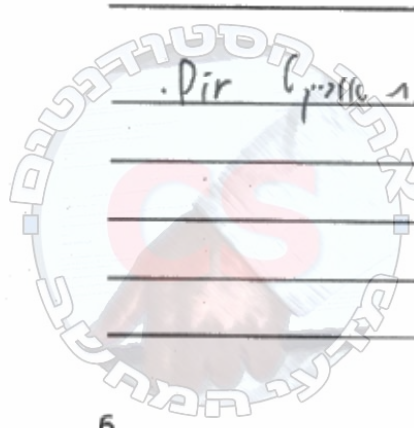
```

1. #include <string>
2. #include <iostream>
3. using namespace std;
4. class File {
5.     inline virtual void print() const { cout << "File name is: " << name; }
6.     protected:
7.     string name;
8.     public:
9.     File(string &s) : name(s) { }
10.    string& getName() const { return name; }
11. };
12. class Dir : public File {
13.     unsigned int dirsize;
14.     public:
15.     inline virtual void print() const { File::print(); cout << "Dir size is: " << dirsize; }
16.     Dir(int size) : dirsize(size) { }
17. };
    
```

שורה: 8 שגיאה: הפונקציה מוגדרת כ-const אך המשתנה name אינו const. תיקון: להוסיף const למשתנה name.
 string getName() const { return name; }

שורה: 11 שגיאה: קונסטרוקטור File::print() מוגדר כ-private (בשורה 5). תיקון: להוסיף protected לקונסטרוקטור.
 File::print() protected

שורה: 12 שגיאה: אין קונסטרוקטור ברירת default ל-Dir. תיקון: להוסיף קונסטרוקטור ברירת default ל-Dir.
 File::File() {}



מה יהיה הפלט של התוכנית הבאה?

```

#include <string>
#include <iostream>
using namespace std;

class Vehicle {
protected:
    unsigned km;
public:
    Vehicle(unsigned k=0) : km(k) { cout<<"hello\n"; }
    ~Vehicle() { cout<<"bye bye\n"; }
    virtual void drive(unsigned k) { km+=k; cout<<km<<endl; }
    void drive() { km++; cout<<km<<endl; }
};

class Car : public Vehicle {
public:
    Car(unsigned k=20) : Vehicle(k) { cout<<"hi\n"; }
    ~Car() { cout<<"chau\n"; }
    void drive() { km *= 2; cout<<km<<endl; }
};

class Truck : public Car{
    unsigned km;
public:
    Truck() : km(100) { cout<<"good morning\n"; }
    void drive(unsigned k) { km = km + 3*k; cout<<km<<endl; }
};

void main() {
    Vehicle* t = new Truck;
    Vehicle* c = new Car(10);
    t->drive(10);
    c->drive();
    delete t;
    delete c;
}

```

פלט:

```

x ka → hello
        good morning
        hello
        hi
        130
        11
        bye bye
x chau
        bye bye

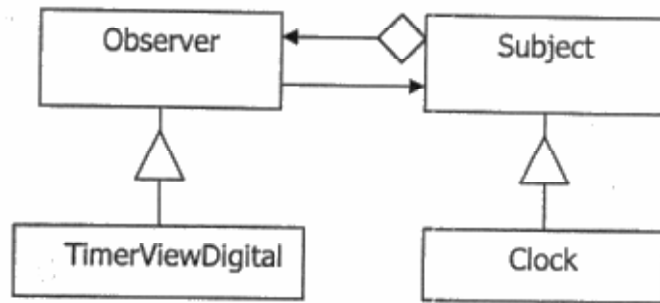
```

-2



שאלה 17: בחלק זה שאלה הפרושה על כמה סעיפים.

17. מוצע התיכון הבא:



- א. Observer הינו אובייקט המעוניין שיקראו לו כשמשהו מעניין קורא.
 - ב. Subject הינו אובייקט שמדווח ל- Observer שנרשם אליו לקבל עדכונים.
 - ג. ל- Subject יכולים להיות הרבה אובייקטי Observer שנרשמו אליו לקבל עדכונים.
 - ד. מהם יורשים
 - i. Clock שהינו בעצם Subject, והוא דואג לדווח למי שנרשם אליו, כל שניה.
 - ii. TimerViewDigital שהינו בעצם Observer שמעוניין שיקראו לו כל שניה.
- הוא מתוכנן לבצע פעולה כלשהי כל X שניות, לכן כשנקרא על עידכון (כל שניה), הוא מבצע פילטור על השניות, וכשמגיע הזמן הרלוונטי מבצע את אשר צריך.
- ה. אובייקטים
 - i. במערכת ישנו אובייקט גלובלי Clock (שהוא בעצם subject), שמתעורר כל שניה, ומדווח על כך לכל הרשומים.
 - ii. אנחנו מייצרים 2 אובייקטים קונקרטיים TimerViewDigital, האחד אמור לדווח כל 2 שניות, והשני כל 3 שניות. 2 האובייקטים נרשמים אל ה- Clock ומקבלים שירותי "הערה" כל שניה.
- ג. דוגמת הרצה: עבור ה- main במסגרת השמאלית יתקבל הפלט המופיע במסגרת הימנית:

```

main
#include "Observer.h"
#include "Subject.h"

int main() {
    //Clk1 is working with 2 secs resolution
    TimerViewDigital clk1("Clk1", 2);
    //Clk2 is working with 3 secs resolution
    TimerViewDigital clk2("Clk2", 3);

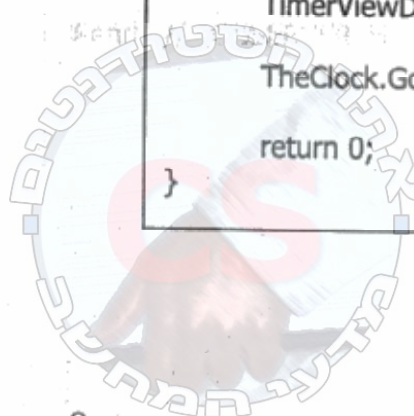
    TheClock.Go();

    return 0;
}
    
```

```

output

Clock at 1 ticks
Clock at 2 ticks
    Clk1 has woken up!
Clock at 3 ticks
    Clk2 has woken up!
Clock at 4 ticks
    Clk1 has woken up!
Clock at 5 ticks
Clock at 6 ticks
    Clk1 has woken up!
    Clk2 has woken up!
Clock at 7 ticks
Clock at 8 ticks
    Clk1 has woken up!
Clock at 9 ticks
    Clk2 has woken up!
Clock at 10 ticks
    Clk1 has woken up!
    
```



Observer.h

```

#ifndef _OBSERVER_H_
#define _OBSERVER_H_

class Subject;

class Observer {
public:
    Observer(Subject& rSubject);
    virtual ~Observer() {}

    virtual void NotificationFromSubject(const Subject & rSubject) = 0;

private:
    Subject m_rSubject;
    void SetSubject(Subject& rSubject);
    void GetSubject(Subject& rSubject) const;
};

class TimerViewDigital : public Observer {
    const char*   m_timerName;
    int           m_wakeUpTime;
public:
    virtual void NotificationFromSubject(const Subject& rSubject) {} ✓
    //ctor
    TimerViewDigital (Subject& rSubject, const char* t, int w) : ✓ ✓
        Observer(rSubject), m_wakeUpTime(w), m_timerName(t) {}
    virtual ~TimerViewDigital ();
};

#endif // _OBSERVER_H_

```

Subject.h

```
#ifndef _SUBJECT_H_
#define _SUBJECT_H_
#include<windows.h>
#include <list>
using namespace std;

class Observer;

class Subject {
    list<Observer*>    m_observers;
public:
    Subject() {}
    virtual ~Subject () {}
    int AddObserver(Observer* pObs);
    int RemoveObserver(Observer* pObs);

int GetObserverCount() const { return m_observers.size(); }

protected:
    void NotifyObservers();
};

class Clock : public Subject {
public:
    Clock() : m_numOfTicks(0) {}
    virtual ~Clock() {}
    void Go() {
        while (1) {
            Sleep(1000); //sleep for 1 second (1000 msec) and then continue
            ++theClock; ++(*this)
        }
    }

    int WhatTime() const { return m_numOfTicks; }

    clock operator ++ ();
};

private:
    int m_numOfTicks;
};

extern Clock theClock;
#endif // _SUBJECT_H_
```

להלן הדרישות מכל מודול

i. Subject תומך בפונקציונליות הבאה

1. הוסף observer - כך שיקבל נוטיפיקציות

2. הסר observer - כך שיפסיק לקבל נוטיפיקציות

3. יידע את כל ה- observers - שהשתנתי, ומי שמעוניין שיעשה משהו בנידון

ii. Clock תומך בפונקציונליות הבאה

1. התחל לעבוד - לספור כל שניה, ולדווח לרשומים

2. מה השעה (כמה ticks עברו מתחילת הספירה)

3. ... משהו שחסר וישאל בשאלות להלן

iii. Observer תומך בפונקציונליות הבאה

1. עדכון הגיע

iv. TimerViewDigital תומך בפונקציונליות הבאה

1. כשעדכון הגיע - בודק אם מס השניות שעברו תואם את הזמנים שעליהם אמור להגיב.

אם כן מדפיס הדפסה (ראה פלט), אם לא מתעלם מההנוטיפיקציה.

17.1 השלימו את הקבצים Subject.h, Observer.h לפי הדרישות הנ"ל.

17.2 צרו את הקבצים Subject.cpp, Observer.cpp לפי הדרישות הנ"ל.



1. ענו על השאלות הבאות המתייחסות לקוד המוצג ולקוד שהושלם על ידכם. שימו לב: חלק מהשאלות מבקשות השלמה נוספת של קוד.

17.3.1 בקובץ `subject.h` והצהרה `class Observer;` - למה צריך אותה ומדוע היא מספיקה?

צריך אותה מכיוון שב `Subject` יש `list` של `Observers`.
`Observer` הוא `user-defined class`, לכן צריך "להגדיר" את ההגדרה שלו.
 יוצרים את המחרטום שלו, היא גמיש קיימת, ואין הוא יכול "לכבוש" אותו.
 ולכן ולא את שום היגיון של יצירת רשימה של `Observers` אליה.

✓

17.3.2 מתי מיוצר האובייקט הגלובלי `theClock`; `Clock`? מי קורא לבנאי שלו?

באובייקט "יוצר" לפני החילוף `main`, ובקובץ `subject.cpp` יש לקיים לבנאי שלו (צג 17).

✓

17.3.3 כשמייצרים את האובייקט הגלובלי `theClock`; `Clock` - מה סדר הקריאות והביצוע (מי נקרא ומה מתבצע) כתוצאה מכך?

ראשית נקרא `ctor` של `clock` - `clock` - `Subject`,
 ואחר כך נקרא `ctor` של `clock` (המיוצר האחרון של `clock` הוא `clock`).
 מה סדר הקריאות והביצוע?
 -1

17.3.4 מה המשמעות של להגדיר מתודה וירטואלית כ- `inline`?

המשמעות היא שהמטודה `inline` נכנסת בגודל הקובץ במקום הקריאה לה.
 כונקלים וירטואליים קורים כי בהחלפה איננו סוקרים מקיפים, אלא אנו מיישמים אותם.
 אחרת נקרא לכל שינויים "קוד", "קוד קטן יותר", איננו סוקרים מקיפים כגילויים ולא איננו
 מודים להם (שיכול להיות אחרת במקרה כגילויים - פולימורפיזם).
 איננו יודעים מה גופי הקוד קיימים - לנדרש מקומותיהם בשל אמצעים `inline`, אלא
 נחשבים שהם קודים קטנים וזמן קטן.

ד"ר אילן סגל
 מרצה במחלקת הנדסת תוכנה ורשתות

17.3.5 מה משמעות ההצהרה ומה ההשלכות שלה
`virtual void NotificationFromSubject() = 0;`

למשך הפלטמה - הנוקליה היא וינלמית טבוים, זה שמיין את החלק Observer
לחלקים אפלטוקליה.
ההשלכות: ו. זה ניתן לילוי אובייקטים גלינס Observer.
2. לחלק פנונוי ה Observer חייג לחמש את פנוקליה הניתל,
אחריו גם היא רהים אפלטוקליה.

17.3.6 מדוע אנו זקוקים ל- virtual Dtor? מתי כן ומתי לא?

אנו זקוקים ל דטור וינלמית גיוון אנו חוליק שבאנו נקרא ה דטור
הוא יקרא לני סוז פדקם מוולבד ולא לני או מאלבד.
גיוון מאלבד לנה חק יכל לאלבד למה למוכיל גמל נוגי (כולי-מכניב),
נלה שבאנו מאלבד נמוס יקרא ה דטור ל מאלבד לו.
אנו זקוקים לכן אם גמל פנונוי מבלד חקלמ מאלבד (חזוטק מחווי) לזכר
למי מאלבד חקלמ ל לחלק חגיסי.

17.3.7 הקוד ++this) במתודה Go() לא עובר קומפילציה. הוסף את האופרטור המתאים במקום הרלוונטי.

הוסף ++this



קוד עבור Observer.cpp

```
void TimerViewDigital::NotificationFromSubject (const Subject & rSubject)
{
if (&rSubject). ✓
    if ( (clock_t)rSubject.whatTime() % m-wakeupTime == 0)
        cout << m-timerName << " has woken up." << endl;
}
```

```
Observer::Observer (Subject & rSubject)
{
    rSubject.addObserver (this);
} ✓
```



Subject.cpp קוד עבור

```
int Subject::AddObserver(Observer* pObs)
{
    m_observers.push-back(pObs); ✓
    return 0;
}
```

!find == ind

```
int Subject::RemoveObserver(Observer* pObs)
{
m_observers.erase(find(m_observers, pObs));
return 0;

```

```
list<Observer*>::iterator it;
for(it = m_observers.begin(); it != m_observers.end(); it++)
{
    if(*it == pObs)
    {
        m_observers.erase(it);
        return 0; ✓
    }
}
return 1;
```

```
void Subject::NotifyObservers()
```

```
{
    list<Observer*>::iterator it;
    for(it = m_observers.begin(); it != m_observers.end(); it++) ✓
        (*it) -> NotificationFromSubject(*this);
}
```




```
clock& Clock::operator++()
```

```
{
```

```
    m_numOfTicks++; cout << "Clock at " << m_numOfTicks << " ticks";
```

```
    NotifyObservers();
```

```
    return *this; return *this;
```

```
}
```

```
clock theClock; ✓
```

בהצלחה

