



תכנות מתקדם 2

מועד ב' תשע"ב

	30 נק'	1
	25 נק'	2
	15 נק'	3
	15 נק'	4
	15 נק'	5
	100 נק'	סה"כ

31.8.12

המחלקה למדעי המחשב

89-211

מרצה:

יריב טל.

מתרגל:

איגור רוכלין.

משך הבחינה:

שלוש שעות. אין הארכה.

חומר עזר:

אסור להכניס כל חומר עזר.

הנחיות כלליות:

רצוי לענות בגוף הבחינה. אחרי כל שאלה יש מקום לתשובות.
אם עניתם על שאלה במחברת – ציינו זאת בגוף הבחינה!
חובה להגיש את המחברת ביחד עם טופס הבחינה בסיום הבחינה.
חובה לענות על השאלות בעברית, אלא אם יש אישור מהדיקן.

הנחיות טכניות:

במידה ונדרשתם לתת נימוק, אזי הוא חובה. כלומר, תשובה לא מנומקת לא תקבל נקודות כלל.
נימוק לא נכון יגרור פסילת השאלה.
בכל השאלות – מספר השורות שניתנו לפתרון אינו מרמז על אורך התשובה.

בהצלחה ☺



שאלה 1: Code Improvements & Testing (30 נקודות)

נתון הקוד הבא:

```
// OptionalValue is used when a function has an optional value.
// Example:
//     time_t getCurrentTime(OptionalValue timezone);
//
//     void printTime() {
//         OptionalValue defaultTimeZone, grinchTimeZone(0);
//         std::cout << "Current time: " << getCurrentTime(defaultTimeZone);
//         std::cout << ", GMT Time: " << getCurrentTime(grinchTimeZone);
//     }
//
class OptionalValue {
public:
    OptionalValue() : hasValue(false) { }
    OptionalValue(int value) : hasValue(true), m_value(value) { }

    // Return true if contains a value, false if not
    bool HasValue() { return hasValue; }

    // Get the stored value. Assumes there *is* a value (use hasValue() first to
    // check if there is a value)
    int getValue() { return m_value; }

    // Assign a value
    void setValue(int VALUE) { hasValue = true; m_value = VALUE; }

    // Set back to no-value
    void clear() { hasValue = false; }

    // operator* is used to access internal value as if OptionalValue is a pointer
    // to int. Note that it assumes that OptionalValue actually has a value.
    // Example usage:
    //     OptionalValue noValue, withValue(5);
    //     *withValue = 7; // ok
    //     *noValue = 3; // undefined behaviour - do not do this!
    int &operator*() { return m_value; }

private:
    bool hasValue;
    int m_value;
};
```

א. (6 נקודות) מיצאו לפחות שתי חריגות מה-Naming Conventions בהם יש שימוש בקוד.



ב. (6 נקודות) אילו פונקציות יכולות להיות מוגדרות כפונקציות const? נמקו ורישמו את החתימה של הפונקציות לאחר הוספת ה-const.

ג. (6 נקודות) בקוד מתועדות הנחות שצריכות להתקיים בעת הקריאה לחלק מהפונקציות. הוסיפו assert יים לקוד הבודקים שההנחות מתקיימות.

ד. נתון הקוד הבא, המשתמש ב-OptionalValue:

```
void foo3()
{
    OptionalValue readwrite(5);
    const OptionalValue readonly(8);

    *readwrite = 7;
    std::cout << *readwrite << std::endl; // ok
    std::cout << *readonly << std::endl; // compiler error
}
```

17. (6 נקודות) הסבירו מדוע הפעלת אופרטור * על המשתנה readonly גוררת שגיאת קומפילציה.

27. (6 נקודות) שנו את OptionalValue כך שהקוד של foo3() יתקמפל ויעבוד. נמקו מדוע הפתרון שלכם עובד!



שאלה 2: Templates & Functors (25 נקודות)

נתון הקוד הבא:

```
template<unsigned int N>
class Magic {
public:
    template <class Func>
    static inline void exec(Func &f){
        f();
        Magic<N-1>::exec(f);
    }
};

template<>
class Magic<0>{
public:
    template <class Func>
    static inline void exec(Func &){
    }
};

void p1() {
    std::cout << "hello" << std::endl;
}
```

א. (5 נקודות) כיתבו אילו אינסטנסיאציות יבצע הקומפיילר ומה יודפס כתוצאה מביצוע שורת הקוד הבאה (נמקו!):
`Magic<0>::exec(p1);`



ב. (5 נקודות) כיתבו אילו אינסטנסיאציות יבצע הקומפיילר ומה יודפס כתוצאה מביצוע שורת הקוד הבאה (נמקו!):
`Magic<1>::exec(p1);`

ג. (5 נקודות) מה מבצע ה-`template?`
רמז: העזרו באינסטנסיאציות הבאות בכדי להגיע למסקנתכם:

```
Magic<2>::exec(p1);  
Magic<3>::exec(p1);
```

ד. (5 נקודות) ממשו את `p1` כ-`Functor` בשם `Print`. הדגימו כיצד תקראו ל-`Magic<2>::exec()` עם ה-`Functor` שכתבתם.

ה. (5 נקודות) שנו את `Print` כך שאפשר יהיה להעביר לו פרמטר (מטיפוס `std::string`) של מה להדפיס. הדגימו קריאה ל-`Magic<2>::exec()` עם ה-`Print` החדש.



שאלה 3: RAII + Templates (15 נקודות)

בתכנות multithreading ו-multithreading ישנם מקרים רבים בהם יש להגן על קטע מסוים כך שרק thread או process אחד יוכלו לגשת אליו בכל פעם. הצורה לבצע זאת היא בעזרת אובייקט נעילה. ישנם הרבה סוגים של אובייקטי נעילה, ברמות שונות של השפעה (ברמת ה-thread או ברמת ה-process), לדוגמא: ThreadsMutex, ProcessMutex, Semaphore. בספרייה שכתבתי לכל האובייקטים המשמשים לנעילה Concept אחד: יש להם פונקציה lock() לביצוע הנעילה ופונקציה unlock() לשיחרור הנעילה. יש לשים לב שכמו שבניהול זיכרון לכל new חייב להיות delete כך גם באובייקטי נעילה על כל lock() חייב להתבצע unlock(). הנה דוגמת קוד: המשתמש ב-ThreadsMutex:

```
void foo(ThreadsMutex &mutex) {  
    bool quickEscape = false;  
  
    mutex.lock();  
  
    // do something that mutex protects  
    // and that might change quickEscape  
    ...  
  
    if (quickEscape) {  
        return;  
    }  
  
    mutex.unlock();  
}
```

א. (2 נקודות) איזה bug ייקרה אם quickEscape יהיה true?

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



שאלה 4: Java (15 נקודות)

חברת טיולים מנהלת נתונים על מדריכים, סוגי טיולים, טיולים בפועל ומטיילים. לכל מדריך יש מספר מזהה, שם וטלפון. החברה מכינה תוכניות לתיאור של טיולים (להלן סוג טיול). לכל סוג טיול יש קוד מזהה, משך (מספר ימים), מכסת משתתפים, מחיר למשתתף ורשימת אתרים שבהם יבקרו. לכל אתר יש שם אתר והוא מצוי בארץ מסוימת. שם האתר ייחודי לארץ שבה הוא מצוי. עבור כל סוג טיול מתכננים את סדר הביקורים באתרים השונים (אתר ראשון, שני וכו'). לכל אתר יש גם משך זמן מומלץ לביקור (בשעות) ודרוג איכות (מס' כוכבים). החברה מבצעת טיולים מכל סוג במשך השנה. כל סוג טיול יכול להתבצע מספר פעמים בשנה אולם בתאריך מסוים יכול לצאת לכל היותר טיול מסוג טיול מסוים. בטיול יכולים להשתתף מספר מדריכים ויש לדעת מי הדריך בכל טיול. כן יש לדעת את מספר המשתתפים.

(4 נקודות) מיהם ישויות (actors) של המערכת?

א. (11 נקודות) צרו תרשים המחלקות של המערכת (class diagram). עבור כל מחלקה, רשום את תכונותיה והקשרים המבניים למחלקות אחרות.

א. (4 נקודות) הסבירו בקצרה מה ההבדלים העיקריים בין DOM ו SAX? מתי עדיף להשתמש בכל אחד מהם?

```
public interface XMLParser {
    public List<Node> getChildNodes(Document document);
}
```



```
public List<Node> getChildNodes(Node node);  
    public Double getChildData(Node node);  
}
```

עליכם לתכנת את המחלקה הבאה ולהשלים את הפונקציה loadTree:

```
public class Tree{  
    .....  
    public static Tree loadTree(XMLParser parser, Document doc){  
        .....  
    }  
}
```

המחלקה מייצגת עץ (לא רק עץ בינארי). הפונקציה loadTree צריכה לקבל XML מהצורה הבאה:

```
<Tree>  
  <data>1.25</data>  
  <Tree>  
    <data>5</data>  
    <Tree>  
      <data>3.25</data>  
      <Tree>  
        <data>4.25</data>  
      </ Tree >  
    </ Tree >  
  <Tree>  
    <data>1.75</data>  
  </ Tree >  
</ Tree >  
<Tree>  
  <data>4</data>  
</ Tree >  
<Tree>  
  <data>2</data>  
  <Tree>  
    <data>1.05</data>  
  </ Tree>  
  <Tree>  
    <data>22</data>  
  </ Tree>  
  <Tree>  
    <data>33</data>  
  </ Tree>  
  <Tree>  
    <data>111</data>  
  </ Tree>  
</ Tree >  
</ Tree >
```

