



תכנות מתקדם 2 – מאזרז' – תשע"ג

89-211-01,02

י"ג אלול תשע"ג, 19/08/13

מרצה: ד"ר מירי בן-ניסן (קופל)

משך הבחינה: 2.5 שעות.

ללא חומר עזר. אין צורך במחברות – רק למי שמבקש לטיוטה.

בהצלחה!

שאלה 1 – Defensive Programming (25 נק')

נתון קטע הקוד המאד לא מוצלח הבא:

```
1. int replaceAndStuff(char buf[], size_t bufs, char* minfc, char* maxfc, char a, char b)
2. {
3.     int i, maxc, minc, repl = 0;
4.     for(i=0; i<bufs; i++) //replace a with b and count replacements
5.         if(buf[i]==a) {buf[i]=b; repl++;}
6.     //find most freq and min freq chars
7.     for(i=0; i<bufs; i++)
8.         if(buf[i]>maxc)
9.             *minfc=buf[i];
10.    for(i=0; i<bufs; i++)
11.        if(buf[i]>minc)
12.            *maxfc=buf[i];
13.    //calculate % of replacements
14.    ratio = repl/bufs*100;
15.    return repl;
16.    return ratio;
17. }
```

א. בתוכנית הזו יש בעיות. חלקן בעיות קומפילציה, חלקן בעיות לוגיות, וחלקן עלולות להתפתח לבעיות בהמשך. רשמו 8 בעיות כאלו בטבלה הבאה, המכילה את שם השורה + הבעיה. אין לחזור על סוג בעיה פעמיים. אם אותה בעיה מופיעה במספר מקומות, רשמו אותה פעם אחת. אני אבדוק רק את ה-8 בעיות הראשונות שתתפרסמו.

(12 נק')

מספר שורה בקוד	הבעיה
1	
2	
3	
4	
5	

ב. כתבו את הפונקציה הנ"ל בצורה הטובה ביותר שתוכלו, לפי כל כללי ה-defensive programming שלמדנו. התוצאה צריכה בסוף להיות פונקציה אחת מרכזית שמבצעת את כל הפעולות שהמקורית מבצעת, וכמובן ניתן להוסיף פונקציות עזר. (נתתי הרבה שורות לא כדי שיהיה לכם ברווח, לא דווקא כי הקוד ארוך).

[illegible]

[illegible]



שאלה 2 – Macro & Template (25 נק')

א. מהו concept של מחלקה? במה הוא שונה מ-interface? נמקו ותנו דוגמה ל-concept. (9 נק')

ב. השימוש ב- Functor בעצם מחליף את הצורך בשימוש ב-dynamic polymorphism (פונקציות וירטואליות וכו'). הסבירו כיצד הוא מחליף זאת, ומדוע הוא נחשב למחליף כזה. (8 נק')

ג. מנו יתרון אחד בשימוש ב- Functor על פני שימוש בפונקציה וירטואלית, ומנו חיסרון אחד של שימוש ב-Functor על פני פונקציה וירטואלית. (6 נק')






שאלה 3 – Android Programming (20 נק')

א. מה ההבדל בין Activity לבין Service? למה אי אפשר להסתדר רק עם Activity? (10 נק')

ב. נתון רצף הפעולות הבאות על מכשיר ה-Android שיכול לתמוך במקסימום 3 תהליכים בו זמנית. רשמתי לכם את תוכן הזיכרון מבחינת התהליכים שרצים בו לאחר הפעולה הראשונה. השלימו את התוכן לגבי כל הפעולות המתוארות להלן (שם התהליך משמאל ולידו את האפליקציה/יות שלו, וכמובן יש לעדכן גם את ה-Activity Manager). אין צורך בציוור האייקונים. (10 נק')

1. מופעל התהליך Home (וה-Activity Manager).

System process Activity Manager	
Home	
Home	

2. מופעלת אפליקצית gmail שמציגה את ה-inbox.



3. מופעל Activity של קריאת הודעה מסוימת.

4. לחיצה על קישור בתוך הודעה שמפעיל את התהליך "דפדפן" המציג דף HTML.

5. לחיצה על "back" כדי לחזור להודעת המייל.

שאלה 4 – נכון/לא נכון (15 נק')

סמנו נכון/לא נכון ונמקו. תשובה לא מנומקת לא תתקבל כלל. תשובה עם נימוק לא נכון גם לא תתקבל.

נימוק	נכון או לא נכון	הסענה	
	נכון / לא נכון	Event Driven Programming הוא תכנות תלוי מערכת הפעלה.	א
	נכון / לא נכון	בהינתן פונקציה בשם <code>rbv()</code> שמחזירה אובייקט <code>Foo</code> ע"י העתקה (by value), מבחינת הקומפיילר, אין הבדל בין הקוד: <code>Foo x = rbv();</code> לבין הקוד: <code>Foo x;</code> <code>x = rbv();</code> שמופיעים ב- <code>main</code> , מבחינת כמות האופטימיזציות שיתבצעו.	ב
	נכון / לא נכון	שימוש ב-DLL בתוך קוד של <code>C++</code> מאפשר קריאה לפונקציות שכתובות בשפת תכנות אחרת ע"י יצירת מצביע אליהן.	ג
	נכון / לא נכון	Debugging הוא אחת הדרכים לשפר את איכות הקוד שלנו.	ד
	נכון / לא נכון	Configuration Management מסתכם בלנהל גרסאות שונות של קבצי הקוד וניהול עבודה במקביל על הקוד.	ה



שאלה 5 – שאלה מהתרגול (15 נקודות)

עבור כל אחד מקטעי הקוד הבאים ציינו את כל האפשרויות לפלטים השונים.

א. נתון JSP הבא:

```
<HTML>
<BODY>
<%
    int number = 5;
%>
Hello! The time is now
<%
    out.println(" "+number);
%>
</BODY>
</HTML>
```

ב.

```
public class TestC extends Thread {
    int id;
    TestC(int id) {
        this.id = id;
        start();
    }
    public static void main(String[] args) {
        Thread t = new Thread(new TestC(11));
        t.start();
    }
    synchronized public void run() {
        id++;
        System.out.print(id);
    }
}
```



ג.

```
public class TestD extends Thread {  
    int i=0;  
    public static void main(String[] args) {  
        TestD test = new TestD();  
        test.start();  
        TestD test1 = new TestD();  
        test1.start();  
        TestD test2 = new TestD();  
        test2.start();  
    }  
    public void run() {  
        i=i+1;  
        System.out.println("Hello World"+i);  
    }  
    public void run1() {  
        i=i+2;  
        System.out.println("Hello World"+i);  
    }  
    public void run2() {  
        i=i+3;  
        System.out.println("Hello World"+i);  
    }  
}
```
