

תכנות מערכות - עבודה הגשה 4

שאלה 1 :

(1) הפלט התוכנית הנתונה :

```
in constructor A
in constructor B
in constructor A
in constructor B
in constructor A
in constructor B
in f1 A
in dtor B
in dtor A
in constructor A
in constructor B
in f2 B
in f1 B
in dtor B
in dtor A
in dtor B
in dtor A
```

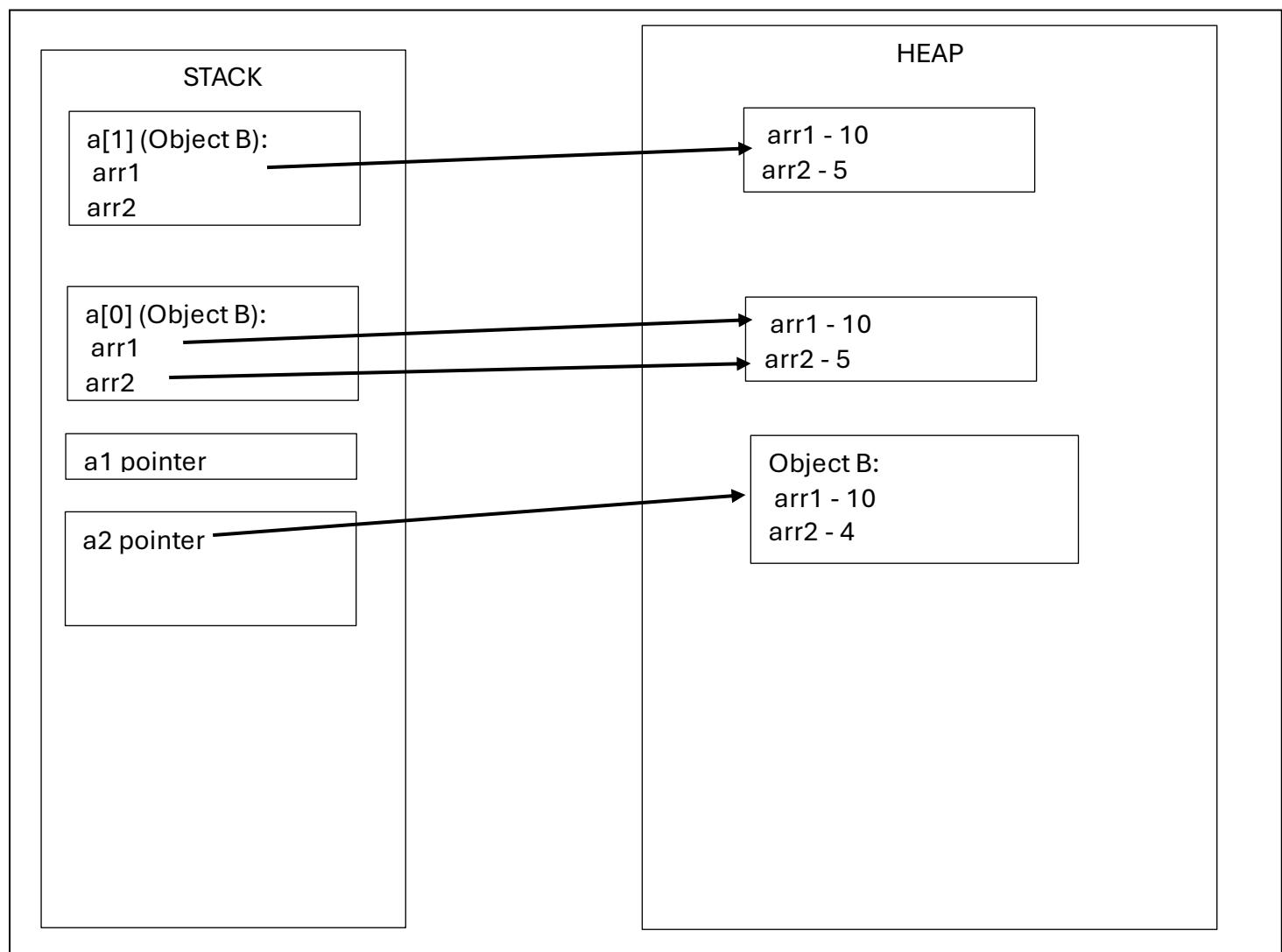
```
int main(){

    B a[2]; →      יצירה של מערך בגודל 2 של אובייקטים מסוג B
    A *a1 = new B(); →      קראיה לבנאים של A ו B::F1
    a1->f1(3); →      הפעלה של פונקציה
    delete a1; →      מחיקת האובייקט הפוך מסדר הייצירה
    A *a2 = new B(); →      יצירה של אובייקט חדש מסוג B
    a2->f2(4); →      F2 וירטואלית , תופעל הפונקציה של B::F2
    a[0].f1(5); →      קראיה לפונקציה F1::B על האובייקט הראשון במערך
    return 0;
}
```

(2) בקוד הנתון יש מספר זליגות זיכרון :

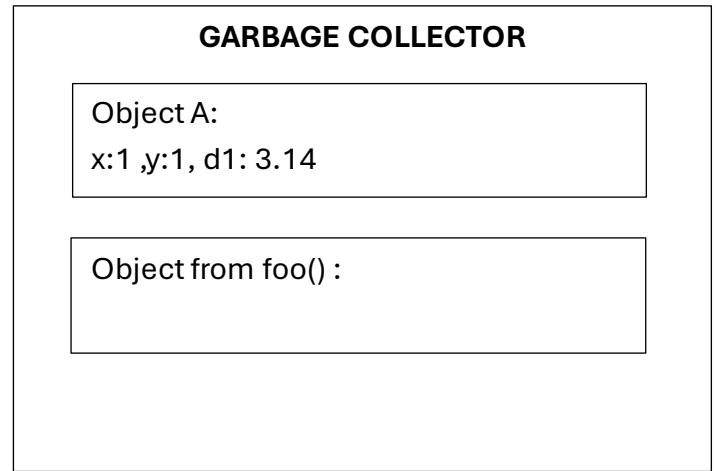
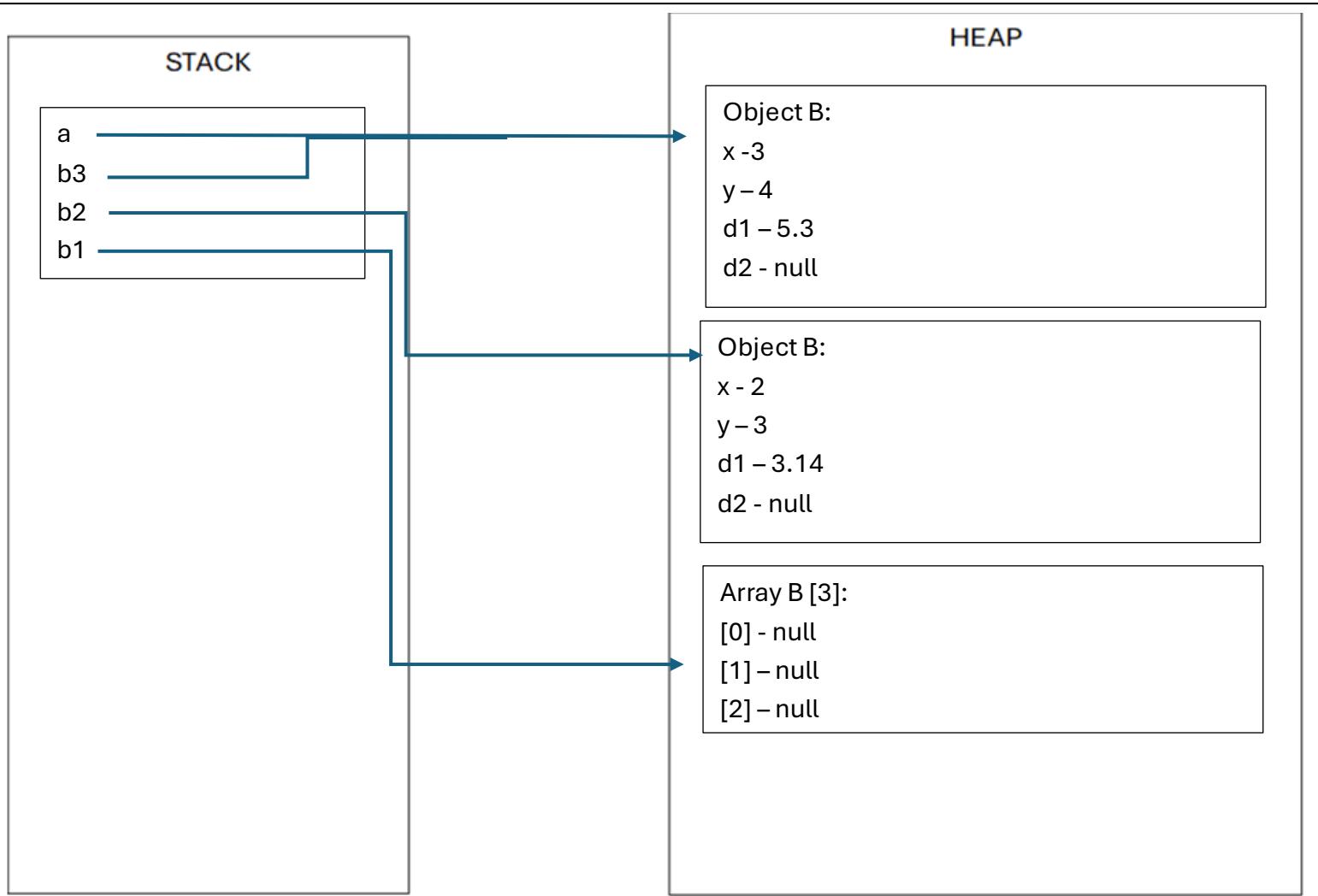
- א. זליגה ב-A::f1 - הפעוקציה מבצעת `arr2 = new int[size]` אם כבר היה זיכרון מוקצה ב-`arr` הכתובת הקודמת נמחקת.
- ב. זליגה ב-f2 ,B::f1 ,B::f1 ,A - ככל ממציאות הקזאה מחדש ל`arr2 / arr1` מבלי לבצע שחרור זיכרון קיים בכל אחד מהם.
- ג. שימוש ב-`new` ב-`a2` אבל לא בוצעה מיחקה עם `delete`.
- ד. ההורס של A משחרר רק את `arr2` וההורס של B משחרר רק את `arr1` , במצב כזה שפונקציה מקצת אובייקט לשנייהם תהיה זליגה כי כל מחלוקת אחריות על מצביע אחד בלבד.

(3) תמונה זיכרון :



שאלה 2 :

(א)



ב) ה garbage collector ימחק את כל האובייקטים ב heap שאין להם מצביע קיים ב stack ,

האובייקטים שימחקו הם :

1. האובייקטים שנוצרו בתוך פונקציה (foo) :

בתוך (a) foo נוצר אובייקט חדש וברגע שהפונקציה הסתיימה המשטנה המקומי a נמחק והאובייקט נשאר ללא מצביע קיים.

2. בתוך ([1] foo , b1(b2 , נוצרו אובייקט () A new ואובייקט () B new גם הם נשארו ללא מצביעים ברגע שהפונקציה הסתיימה (כי 1 או 2-a-היו משתנים מקומיים).

3. אובייקט ה A-הראשון שנוצר בשורה הראשונה של ה main-ברגע שביצעו a = b3 בשורה האחרונה, איבדנו את המצביע היחיד שהוא לאובייקט ה A-המקורי.

שאלה 4 :

תהליך ה – Verification בשלב ה – Linking :

שלב זה הוא חלק מהתהליך ה – Linking , ומטרתו העיקרית היא להבטיח שקובץ ה – ByteCode שנטען הוא בטוח להרצה והוא מפרק את כלל ה- JVM.

אחר שקובץ מחלוקת יכול להציגו מקורות לא ידועים ה- JVM אינה סומכת על הקוד ובמצעת סדרת בדיקות :

1. **בדיקה תקינות מבנית** - ידוא שהקובץ בניו בפורמט נכון של JAVA ושהכל במבנה חוקי.

2. **אימות ByteCode** - זהו החלק המורכב ביותר, בו המערכת עוברת על פקודות הקוד ומבודדת שהן "גיאניות". לדוגמה:

◦ אין חריגה ממחסנית (Stack Overflow/Underflow).

◦ שיטוגי הנתונים מתאימים לפעולות המבוצעות עליהם.

◦iscal פקודות קפיצה בתחום הקוד מובילה לכתובת חוקית בתחום אותה מתודה.

3. **בדיקה הרשות** - ידוא שהקוד לא מנסה לבצע פעולות אסורות, כמו גישה לשדות פרטיים של מחלוקת אחרת או ירושה מחלוקת שהוגדרה כ- final .

4. **אתחול משתנים** - ידוא שכל משתנה מקומי מקבל ערך לפני שימושים בו בפועל.