

לפניכם כותרות המחלקות BB, CC והתכונות שלהן:

```
public class BB
{
    private int a;
    protected int b;

    ****
}

public class CC extends BB
{
    private int c;

    ****
}
```

הערה: שימו לב – לשתי המחלקות אין פעולות `get` ו-`set`.  
לפניכם המחלקה `Tester` הכוללת פעולה ראשית:

```
public class Tester{
    public static void main(String[] args)
    {
        BB [] b = new BB[8];
        b[0] = new BB();
        b[1] = new BB(16);
        b[2] = new BB(b[0]);
        b[3] = new BB(b[1]);
        b[4] = new CC(9);
        b[5] = new CC(5);
        b[6] = new CC(b[0]);
        b[7] = new CC(b[1]);
    }
}
```

א. לפניכם תרשים עצמים שנוצרו בעקבות הרצת קטע הקוד.  
כתבו במחלקות `BB` ו-`CC` את הפעולות הבונות הנדרשות כדי לקבל את העצמים שבתרשים.  
ציינו בעבור כל אחד מן העצמים שבתרשים את הפעולה הבונה על פיה הוא נוצר.  
הערה: אין להוסיף פעולות שאינן בונות במחלקות `BB` ו-`CC`. פתרון הכולל פעולות שאינן בונות לא יזוכה בנקודות.

(שימו לב: המשך השאלה בעמוד הבא.)

0	1	2	3	4	5	6	7
BB	BB	BB	BB	CC	CC	CC	CC
a= 10 b= 1	a= 10 b= 16	a= 10 b= 1	a= 10 b= 16	a= 10 b= 9 c= 7	a= 10 b= 5 c= 7	a= 10 b= 1 c= 19	a= 10 b= 16 c= 19

ב. נתון קטע קוד תקין והפלט שהתקבל בעקבות הרצתו. כמו כן, נתון קטע קוד עבורו התקבלה שגיאת הידור (קומפילציה).

ממשו את הפעולות goo ו-foo במחלקות המתאימות (BB ו-CC) על פי עקרונות תכנות מונחה עצמים, כך שיתקבל הפלט שלפניכם (יש לממש במחלקות רק את הפעולות הנחוצות כדי שהקוד ירוץ).

קטע קוד תקין:

```
b[0].foo(0); //1
b[0].foo(3); //2
((CC)b[4]).foo(); //3
((CC)b[6]).foo(); //4
((CC)b[5]).goo(); //5
((CC)b[7]).goo(); //6
```

הפלט שהתקבל עבור הרצתו:

```
BB foo: 11 //1
BB foo: 14 //2
CC foo: 7 //3
BB foo: 26
CC foo: 19 //4
BB foo: 30
CC goo //5
CC goo //6
```

כמו כן, שתי הפעולות בקטע הקוד שלפניכם אינן תקינות ומתקבלת שגיאת הידור (קומפילציה).

```
b[3].foo();
b[2].goo();
```