

מספר סידורי: _____ מספר ת"ז: _____

בחינה בתוכנה 1

סמסטר ב' תשע"ז, מועד א', 24 ביולי 2017
סיון טולדו, ברית יונגמן, לנה דנקין

משך הבחינה שלוש שעות.

יש להניח שהקוד שמופיע במבחן מתאים לגירסה 8 של ג'אווה, אלא אם כן צויין אחרת בשאלה. כמו כן, ניתן להניח שכל החבילות הדרושות יובאו, ואין צורך לכתוב שורות import.

בבחינה זו מופיע קוד שבחלקו אינו מתקמפל, אינו רץ או שנוגד את הסטנדרטים של java כפי שנלמדו בקורס, וזאת מתוך מטרה לבחון ידע והבנה של נושאים מסויימים. אין לראות בקטעי קוד אלה דוגמא לכתיבה נכונה ב java.

יש לסמן את התשובה הטובה ביותר בתשובון. לא יינתן ניקוד על סימון תשובה בטופס הבחינה, במחברת הבחינה, או בטופס הנימוקים.

יש לנמק את התשובות המצריכות נימוק בטופס הנימוקים המצורף בלבד. נימוק חסר או לא נכון עלול לגרום לאי קבלת נקודות על שאלה גם אם התשובה המסומנת היא הטובה ביותר. המקום המיועד לנימוקים בטופס מספיק לתשובות תמציתיות.

יש לצרף את טופס המבחן למחברת הבחינה. מחברת ללא טופס עזר תפסל. תשובות ונימוקים במחברת הבחינה לא יבדקו.

יש למלא מספר סידורי (מספר מחברת) ומספר ת"ז על כל דף של טופס הבחינה.

אסור השימוש בחומר עזר כלשהו, כולל מחשבוניו או כל מכשיר אחר פרט לעט.

לטופס הבחינה מצורף דף לשאלות הסטודנטים. יש לכתוב שאלות שמתעוררות במהלך הבחינה בדף זה ולהעביר לסגל הקורס. שאלות ענייניות תענינה על ידי סגל הקורס בפני כל הנבחנות/ים.

© כל הזכויות שמורות למחברים. מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן במאגר מידע, בכל דרך שהיא, בין מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

בהצלחה!

1. שאלה 1 (5 נק'):

מה יודפס בהרצת הפונקציה הבאה?

```
1 public static void foo() {  
2     int x = 0;  
3     try {  
4         if (x==0)  
5             throw new RuntimeException() {  
6                 public void bar() { System.out.printf("bar\n"); }  
7             };  
8         x += 1;  
9     } catch (Exception e) {  
10        x += 2;  
11    } finally {  
12        x += 7;  
13    }  
14    System.out.printf("%d\n",x);  
15 }
```

- א. 1
- ב. 9
- ג. הפונקציה תזרוק חריג ולא תדפיס שום דבר.
- ד. 10
- ה. 0
- ו. 3
- ז. הקוד לא מתקמפל בגלל שגיאת קומפילציה בשורה אחת (יש לנמק איזה ולמה)
- ח. הקוד לא מתקמפל בגלל שגיאת קומפילציה ביותר משורה אחת (יש לנמק איזו ולמה)
- ט. 8
- י. 2

2. שאלה 2 (5 נק'):

מה יקרה בהרצת הפונקציה הבאה?

```
1 public static void main() {  
2     java.util.Set<? extends Exception> s = null;  
3     Exception e1 = null;  
4     Exception e2 = null;  
5     s.add(e1);  
6     e2 = s.iterator().next();  
7 }
```

- א. הקוד לא יתקמפל בגלל שגיאת קומפילציה בשורה 6
- ב. הקוד לא יתקמפל בגלל שגיאת קומפילציה בשורה 5
- ג. הקוד לא יתקמפל בגלל שגיאות קומפילציה בשורות 1, 6
- ד. הקוד יתקמפל, ירוץ, וייצר חריג מסוג NullPointerException בשורה 6
- ה. הקוד לא יתקמפל בגלל שגיאות קומפילציה בשורות 1, 5
- ו. הקוד לא יתקמפל בגלל שגיאות קומפילציה בשורות 5, 6
- ז. הקוד לא יתקמפל בגלל שגיאות קומפילציה בשורות 1, 5, 6
- ח. הקוד יתקמפל, ירוץ, וייצר חריג מסוג NullPointerException בשורה 5
- ט. הקוד לא יתקמפל בגלל שגיאת קומפילציה בשורה 1

3. שאלה 3 (5 נק'):

מה יקרה בהרצת הקוד הבא?

```
public class ExamQuestion {  
  
    public static class Base {  
        private void test() { System.out.print('B'); }  
    }  
  
    public static class Sub extends Base {  
        private void test() { System.out.print('S'); }  
    }  
  
    public static void main(String[] args) {  
        Base x = new Base();  
        x.test();  
        Sub y = new Sub();  
        y.test();  
        Base z = new Sub();  
        z.test();  
        Sub w = (Sub) new Base();  
        w.test();  
    }  
}
```

- א. ידפיס BSBB
- ב. ידפיס BSBS
- ג. ידפיס BSSB
- ד. הקוד לא מתקמפל בכלל.
- ה. ידפיס BBSB
- ו. ידפיס BBSS
- ז. ידפיס BSS ויעוף על חריג.
- ח. ידפיס BBS ויעוף על חריג.
- ט. ידפיס BBB ויעוף על חריג.
- י. ידפיס BSB ויעוף על חריג.

4. שאלה 4 (5 נק'):

```
public class DynamicGraph {  
  
    public class Vertex {  
        private final int index;  
        public Vertex() {  
            ... // not shown in the exam  
        }  
    }  
  
    private boolean[][] representation = new boolean[0][0];  
  
    public DynamicGraph() {  
    }  
  
    public void addEdge(Vertex u, Vertex v) {}  
    ... // more methods, not shown in the exam  
}
```

נתונה הגדרה חלקית של מחלקה שמייצגת גרף לא מכוון $G=(V,E)$ שניתן להוסיף לו ולהוריד ממנו קשתות.

לקוחות מוסיפים לגרף צמתים על ידי יצירה של עצמים מטיפוס `DynamicGraph.Vertex` ומוסיפים קשתות על ידי קריאה לשירות `addEdge` (מתודה). צמתים מיוצגים בגרף על ידי מחלקה פנימית עם שדה בודד `index` מטיפוס `int`. לצמתים שונים יש אינדקסים שונים, כולם אי שליליים. הגרף מיוצג על ידי שדה בודד מטיפוס מערך דו-מימדי של ערכים בולאניים. פונקציית ההפשטה (*abstraction function*) ממפה את המערך לגרף שבו הקשת u,v קיימת אם ורק אם `representation[i][j]==true` עבור i שהוא האינדקס של u ו- j שהוא האינדקס של v .

השאלה מתייחסת לטענות הבאות:

1. החוזה של המחלקה `DynamicGraph` חייב לכלול משתמר ייצוג לא טריויאלי (שאינו תנאי ריק).
2. לשירות `addEdge` חייב להיות תנאי קדם שכולל תנאים מעבר לתנאים `u!=null, v!=null` שאותם נכליל בחוזה בכל מקרה.
3. את הבנאי של `Vertex` אפשר להגדיר בצורה נכונה בין אם המחלקה הפנימית `Vertex` מוגדרת כמחלקה סטטית או כמחלקה לא סטטית (כלומר בין אם נוסיף להגדרה שלה את המילה `static` או נשאיר אותה כפי שהיא).

- א. טענות 2, 3 בלבד נכונות.
- ב. טענה 3 בלבד נכונה.
- ג. אף טענה לא נכונה.
- ד. טענה 1 בלבד נכונה.
- ה. טענה 2 בלבד נכונה.
- ו. טענות 1, 2 בלבד נכונות.
- ז. טענות 1, 3 בלבד נכונות.
- ח. כל הטענות נכונות.

5. שאלה 5 (5 נק')

```
public abstract class A {  
    public A() {...}  
    ... // more code, not shown  
}  
  
public class C extends A {  
    public C() {  
        i = 7;  
        j = this.method(i);  
    }  
}
```

בחר/י את הטענה הטובה ביותר:

- א. עבור כל השלמה של הקוד החסר במחלקה A הקוד לא מתקמפל משום שגם המתודה method וגם המזהים i, j, אינם מוגדרים.
- ב. יש השלמה של הקוד החסר כך שהקוד מתקמפל.
- ג. עבור כל השלמה של הקוד החסר במחלקה A הקוד לא מתקמפל משום שהמתודה method אינה מוגדרת במחלקה C.
- ד. עבור כל השלמה של הקוד החסר במחלקה A הקוד לא מתקמפל משום שחסרה הגדרה למזהים i, j, שמשתמשים בהם בבנאי של C.

6. שאלה 6 (5 נק'):

שאלה זו נוגעת למחלקה BufferedWriter. לפניכם הטענות הבאות:

- טענה 1: בעת ביצוע פעולת close על אובייקט מטיפוס BufferedWriter, תיתכן כתיבה לקובץ.
- טענה 2: ללא שימוש ב BufferedWriter, לא ניתן לכתוב טקסט לקבצים.
- טענה 3: על מנת להקטין את מספר התווים שנכתבים בכל גישה לקובץ, עלינו להקטין את ה Buffer.

בחר/י את התשובה הטובה ביותר:

- א. כל הטענות נכונות
- ב. רק טענות 1+2 נכונות
- ג. רק טענות 1+3 נכונות
- ד. רק טענה 3 נכונה
- ה. רק טענה 1 נכונה
- ו. רק טענות 2+3 נכונות
- ז. רק טענה 2 נכונה
- ח. אף טענה אינה נכונה.

7. שאלה 7 (5 נק')

המחלקה Number ב-java.lang היא מחלקה אבסטרקטית שמרחיבות אותה המחלקות Integer, Double, וכו'. האם בקוד הבא יש שגיאות קומפילציה, ואם כן איפה? סיוע: השפה ממירה מ-float, int, long ל-double, וכן מ-int ל-long בלי שיש צורך באופרטור .cast.

```
public class AnonymousNumber {  
  
    private int privatefield;  
    public int publicfield;  
    public static int staticfield;  
  
    public Number getANumber() {  
        final int variable = 0;  
        return new Number() {  
            @Override public double doubleValue() {  
                return variable; // ***  
            }  
  
            @Override public float floatValue() {  
                return privatefield; // $$$  
            }  
  
            @Override public int intValue() {  
                return 0;  
            }  
  
            @Override public long longValue() {  
                return staticfield; // @@@  
            }  
        };  
    }  
}
```

- א. הקוד מתקמפל.
- ב. בשורה \$\$\$
- ג. בשורות ***, @@@
- ד. בשורה @@@
- ה. בשורה ***
- ו. בשורות ***, @@@, \$\$\$
- ז. בשורות ***, \$\$\$
- ח. בשורות @@@, \$\$\$

8. שאלה 8 (5 נק')

```
1 public class C {  
2  
3     int i;  
4     static int j;  
5  
6     void nonStatic() {  
7         System.out.println(i);  
8         System.out.println(j);  
9     }  
10  
11     static void isStatic() {  
12         System.out.println(i);  
13         System.out.println(j);  
14     }  
15  
16     public static void main(String[] args) {  
17         int k;  
18         C x = new C();  
19         System.out.println(i);  
20         System.out.println(j);  
21         System.out.println(k);  
22         System.out.println(x.i);  
23         System.out.println(x.j);  
24     }  
25 }
```

- א. שגיאת קומפילציה בשורה 12
- ב. שגיאת קומפילציה בשורות 12, 19
- ג. שגיאות קומפילציה בשורות 8, 12, 19, 21
- ד. שגיאות קומפילציה בשורות 8, 12, 21
- ה. אין שגיאות קומפילציה בקוד.
- ו. שגיאת קומפילציה בשורה 19
- ז. שגיאות קומפילציה בשורות 8, 12, 19
- ח. שגיאות קומפילציה בשורות 8, 19, 21
- ט. שגיאות קומפילציה בשורות 12, 19, 21
- י. שגיאות קומפילציה בשורות 19, 21

9. שאלה 9 (5 נק'):

```
public class C {  
  
    public interface I {  
        public int foo(int i); // ***  
    }  
  
    public interface J extends I {  
        public double foo(int i); // $$$  
    }  
  
    public static void bar(I i,int j) {  
        System.out.println(i.foo(j)); // @@@  
    }  
  
    public static void main(String[] args) {  
        bar( (x) -> x+1 , 4); // ###  
    }  
}
```

- א. הקוד מתקמפל ומדפיס 5
- ב. לא מתקמפל בגלל שורות \$\$\$,###
- ג. לא מתקמפל בגלל שורה ### ואם נסיר אותה זה יתקמפל
- ד. לא מתקמפל בגלל שורות @@@ ,###
- ה. לא מתקמפל בגלל שורה \$\$\$ ואם נסיר אותה זה יתקמפל
- ו. הקוד מתקמפל ומדפיס 4
- ז. הקוד מתקמפל ומדפיס 3
- ח. לא מתקמפל בגלל שורה @@@ ואם נסיר אותה זה יתקמפל

10. שאלה 10 (5 נק'):

האם בקוד הבא יש שגיאות קומפילציה, ואם כן, באיזה מהשורות המסומנות?

```
public class EasyGenerics {  
    void msn(java.util.Set    <Number> x) {}  
    void mhn(java.util.HashSet<Number> x) {}  
    void msf(java.util.Set    <Float> x) {}  
    void mhf(java.util.HashSet<Float> x) {}  
  
    void m() {  
        java.util.Set    <Number> fsn = null;  
        java.util.HashSet<Number> fhn = null;  
        java.util.Set    <Float> fsf = null;  
        java.util.HashSet<Float> fhf = null;  
  
        msn(fsfs); // ***  
        msn(fhn); // @@@  
        msf(fhf); // $$$  
    }  
}
```

- א. בשורות **\$, \$\$\$
- ב. בשורה @@@
- ג. בשורות **\$, @@@
- ד. בשורה \$\$\$
- ה. בשורות **\$, @@@, \$\$\$
- ו. הקוד מתקמפל.
- ז. בשורה ***
- ח. בשורות @@@, \$\$\$

11. שאלה 11 (5 נק'):

- נתונה חבילה בג'אווה בשם `com.zechuchit` שמממשת מחלקות וממשקים לייצוג סוגי זכוכית שונים (רגילה, חזקה במיוחד, צבעונית) ועיבודים שונים של זכוכיות (חיתוך, הדפסה, תוספת מדבקות, וכו'). החבילה בשימוש נרחב בתוכנות שונות.
- כרגע יש בחבילה ממשק בשם `Glass` ומחלקה `RegularGlass` שמממשת אותו, ומחלקות `StrongGlass` ו-`ColoredGlass` שמרחיבות אותה. בממשק מוגדרים שירותי מופע `cut` ו-`printOn` שמייצגים עיבודים שונים. השירות `cut` משנה את המימדים של חתיכת זכוכית; המימדים הם חלק מהייצוג של `Glass`.
- לאחרונה צץ בשוק הזכוכיות סוג זכוכית חדש, יצוקה, שלא ניתן לחתוך אותה לאחר שנוצקה. התוכניתניות שמתחזקות את חבילת התוכנה רוצות להוסיף לה מחלקה חדשה `CastGlass` שתציג את הסוג החדש.
- בחר/י עבורן את האפשרות הטובה ביותר.
- הערה: בשאלה הזו תתקבל יותר מתשובה אחת, בהינתן נימוק מתאים.
- א. כדאי שהמחלקה החדשה תרחיב את `Glass` ותדרוס את `cut` עם שירות שלא עושה דבר אלא משאיר את המימדים של הזכוכית בגודל המקורי.
 - ב. כדאי שהמחלקה החדשה תממש ישירות ממשק חדש שיצהיר רק על השירות `printOn` ושהממשק `Glass` ישונה כך שירחיב את הממשק החדש ויוסיף את `cut`, וכמו כן נשנה את `RegularGlass` כך שתרחיב את `CastGlass`.
 - ג. כדאי שהמחלקה החדשה תממש ישירות ממשק חדש שיצהיר רק על השירות `printOn` ושהממשק `Glass` ישונה כך שירחיב את הממשק החדש ויוסיף את `cut`, ואת המחלקות הקיימות לא נשנה.
 - ד. כדאי שהמחלקה החדשה תרחיב את `Glass` ותדרוס את `cut` עם שירות שמחזיר `Exception` שמרחיב את `RuntimeException`.
 - ה. כדאי שהמחלקה החדשה תרחיב את `Glass` ותדרוס את `cut` עם שירות שמחזיר `Exception` שאינו `RuntimeException`.

12. שאלה 12 (5 נק'):

```
public class C {  
  
    private long[] array = new long[ 100000 ];  
  
    @Override public void finalize() {    // ***  
        list.add(this);                  // ###  
    }  
  
    private static LinkedList<C> list = new LinkedList<>();  
  
    public static void main(String[] args) {  
        while (true) {  
            new C(); // $$$  
        }  
    }  
}
```

- א. הקוד מתקפל, רץ אבל עף על `StackOverflowError`, אבל אם נסיר את שורה `###`
הוא ירוץ בלולאה אינסופית בלי לעוף
- ב. הקוד לא מתקמפל – שגיאת קומפילציה בשורות `***,###`
- ג. הקוד לא מתקמפל – שגיאת קומפילציה בשורה `$$$`
- ד. הקוד מתקמפל ורץ בלולאה אין סופית
- ה. הקוד לא מתקמפל – שגיאת קומפילציה בשורה `###`
- ו. הקוד מתקמפל, רץ אבל עף על `OutOfMemoryError`, אבל אם נסיר את שורה `###`
הוא ירוץ בלולאה אינסופית בלי לעוף
- ז. הקוד לא מתקמפל – שגיאת קומפילציה בשורות `***,$$$`
- ח. הקוד לא מתקמפל – שגיאת קומפילציה בשורות `###,$$$`
- ט. הקוד לא מתקמפל – שגיאת קומפילציה בשורה `***`

13. שאלה 13 (5 נק'):

בחן/י את הקוד הבא ובחר/י את התשובה הטובה ביותר. יש להניח כי כל מחלקה מוגדרת

```
public class A {  
    private int i;  
  
    public A(int i) {  
        this.i = i;  
    }  
}  
  
public class B extends A {  
    public B() {  
        this(0); /*  
  
    public B(int i) {  
        super(i);/**  
    }  
}
```

- א. ישנה שגיאת קומפילציה בשורה המסומנת ב-*
- ב. ישנה שגיאת קומפילציה, אך אם נוסיף את הבנאי שלא מקבל ארגומנטים למחלקה A לא תהיה שגיאת קומפילציה.
- ג. ישנה שגיאת קומפילציה בשורה המסומנת ב-**
- ד. הקוד מתקמפל.
- ה. ישנה שגיאת קומפילציה בשורה שמסומנת ב-*, אך אם נשנה את השורה ל `super()`, שגיאת הקומפילציה תיפתר.

14. שאלה 14 (5 נק'):

בחן/י את הקוד הבא ובחר/י את התשובה הטובה ביותר:

```
public class A {  
    private int i = 1;  
    public int j = 2;  
  
    public class C {  
        private int i = 0;  
        public int j = 0;  
    }  
}
```

- א. הקוד לא מתקמפל, גם במידה והמחלקה C הייתה מוגדרת סטטית
- ב. הקוד מתקמפל גם במידה והמחלקה C הייתה מוגדרת כמחלקה סטטית
- ג. הקוד מתקמפל, אך אם המחלקה C הייתה מוגדרת כמחלקה סטטית הוא לא היה מתקמפל
- ד. הקוד מתקמפל, אך במידה והמחלקה C הייתה מוגדרת כמחלקה סטטית הוא לא היה מתקמפל, אלא אם היינו מגדירים את השדות של המחלקה C כשדות סטטים
- ה. הקוד לא מתקמפל, אך אם נשנה את המחלקה C למחלקה סטטית הוא יתקמפל

15. שאלה 15 (5 נק'):

בחנ' את הקוד הבא ובחר' את התשובה הנכונה ביותר. השורה המסומנת ב-*** היא שורת

```
public class Test {
    public enum Operation {
        PLUS, MINUS, TIMES, DIVIDE;
        double calculate(double x, double y) { /**
            double ans = 0;
            switch (this) {
                case PLUS:
                    ans= x + y;
                    break;
                case MINUS:
                    ans= x - y;
                case TIMES:
                    ans= x * y;
                case DIVIDE:
                    ans= x / y;
                default:
                    throw new RuntimeException("No Such Operation: "+this);
            }
            return ans;
        }
    }
    public static void main(String[] args)
    {
        System.out.println(Operation.PLUS.calculate(1, 2)); /**
        //System.out.println(Operation.TIMES.calculate(1, 2)); /**
    }
}
```

- א. כאשר ננסה להריץ את התכנית, נקבל שגיאת זמן ריצה: "No Such Operation: PLUS"
- ב. כאשר ננסה להריץ את התכנית, התכנית תרוץ ותדפיס למסך 3.0
- ג. אף תשובה אינה נכונה
- ד. כאשר ננסה להריץ את התכנית, התכנית תרוץ ותדפיס 0.5
- ה. התכנית לא מתקמפלת – שגיאת קומפילציה בשורה המסומנת ב-***

16. שאלה 16 (5 נק'):

שאלה זו נוגעת לקוד שהוצג בשאלה הקודמת. הניחו כי כעת השורה המסומנת ב-*** כבר אינה בהערה, אך השורה המסומנת ב-** מסומנת בהערה.

- א. התכנית לא מתקמפלת – שגיאת קומפילציה בשורה המסומנת ב-***
- ב. כאשר ננסה להריץ את התכנית, התכנית תרוץ ותדפיס 0.5
- ג. כאשר ננסה להריץ את התכנית, נקבל שגיאת זמן ריצה: "No Such Operation: TIMES"
- ד. כאשר ננסה להריץ את התכנית, התכנית תרוץ ותדפיס למסך 2.0
- ה. אף תשובה אינה נכונה

17. שאלה 17 (5 נק'):

```
public class A {  
    static class B  
    {  
        Integer i;  
        Integer j;  
        B(int i, int j)  
        {  
            this.i = i;  
            this.j = j;  
        }  
    }  
    public static void main(String[] args)  
    {  
        Set<B> set = new HashSet<B>();  
        set.add(new B(1,2));  
        set.add(new B(1,2));  
        set.add(new B(1,3));  
  
        System.out.println(set.size());  
    }  
}
```

- א. בכל ריצה אפשרית, המתודה תדפיס 3
ב. הקוד אינו מתקמפל כיוון שהמחלקה B אינה מממשת את הממשק Comparable
ג. בכל ריצה אפשרית, המתודה תדפיס 2
ד. הקוד אינו מתקמפל כיוון שהמחלקה B אינה דורסת את המתודה hashCode
ה. בכל ריצה אפשרית, המתודה תדפיס 1
ו. הקוד אינו מתקמפל כיוון שלא ניתן ליצור מופע של המחלקה B ללא מופע של המחלקה A.
ז. הקוד אינו מתקמפל כיוון שהמחלקה B אינה דורסת את המתודה equals

18. שאלה 18 (5 נק').

השאלה מתייחסת לממשקים ולמחלקות שהוגדרו בתרגיל 7. מה ידפיס הקוד הבא, שמשמש באותם ממשקים ומחלקות?

```
IPAddress y = IPAddressFactory.createAddress  
    (new short[] { 0x17, 0x18, 0x19, 0x1A });  
System.out.printf("%d",y.getOctet(1));
```

- א. הקוד הזה לא תקין בכלל בהקשר של תרגיל 7 (לא תואם להוראות התרגיל).
ב. ידפיס 17.
ג. ידפיס 23.
ד. ידפיס 18.
ה. ידפיס 24.

19. שאלה 19 (5 נק'):

```
class Box {
    private int x;

    public Box(int x) {
        this.x = x;
    }

    public boolean equals(Object obj) {
        Box other = (Box) obj;
        return x == other.x;
    }
}

public class Test {
    public static void main(String[] args){
        Box b1 = new Box(2);
        Box b2 = new Box(2);
        Box b3 = b2;
        func(b2);
        System.out.print(b1.equals(b2) + ",");
        System.out.print(b3.equals(b2));
    }

    public static void func(Box b){
        ... //implementation is not provided
    }
}
```

לפניכם מספר טענות הנוגעות לשתי המחלקות Box ו Test. שתי המחלקות ממומשות בקובצי java נפרדים.

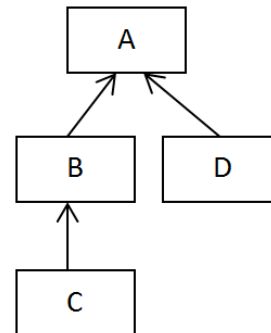
- טענה 1: יתכן מימוש של func שיגרום להדפס false,true בהרצת Test.
טענה 2: יתכן מימוש של func שיגרום לקוד להדפיס false,false בהרצת Test.
טענה 3: יתכן מימוש של func שיגרום לקוד להדפיס true,false בהרצת Test.

בחר/י את התשובה הטובה ביותר, תוך הנחה שהמימוש של Box נותר ללא שינוי.

- א. רק טענות 2+3 נכונות
- ב. אף טענה אינה נכונה.
- ג. כל הטענות נכונות
- ד. רק טענה 1 נכונה
- ה. רק טענה 3 נכונה
- ו. רק טענות 1+2 נכונות
- ז. רק טענות 1+3 נכונות
- ח. רק טענה 2 נכונה

20. שאלה 20 (5 נק'):

```
public class A {  
}  
  
public class B extends A {  
}  
  
public class D extends A {  
}  
  
public class C extends B {  
    public void func(B b) {  
        A a = new A();  
        A aa = (A)b; //1  
        C cc = (C)a; //2  
        D dd = (D)b; //3  
    }  
}
```



בחנ'י את השירות func ובחר'י את התשובה הטובה ביותר:

- א. שורה 3 לא תתקמפל.
- ב. שורה 3 תתקמפל ותזרוק בודאאות ClassCastException בזמן ריצה.
- ג. שורה 1 תתקמפל אך עלולה לזרוק ClassCastException בזמן ריצה.
- ד. שורה 3 תתקמפל אך עלולה לזרוק ClassCastException בזמן ריצה.
- ה. שורה 2 תתקמפל ותרוץ ללא שגיאת ClassCastException.

21. שאלה 21 (5 נק'):

```
class A {  
    public static void foo() {  
        System.out.print("A.foo() ");  
    }  
  
    public void bar() {  
        foo();  
    }  
}  
  
public class B extends A {  
    public static void foo() {  
        System.out.print("B.foo() ");  
    }  
  
    public static void main(String[] args) {  
        B b = new B();  
        b.bar();  
        foo();  
    }  
}
```

מה יודפס בהרצת הקוד הבא? בחר/י את התשובה הטובה ביותר:

- א. A.foo A.foo
- ב. A.foo B.foo
- ג. B.foo A.foo
- ד. B.foo B.foo
- ה. הקוד לא יתקמפל