Compilation vs. Runtime Errors

Syntax error on token "Class", class expected

```
Class MyClass {
    void f() {
        int n=10;
        void g() {
            int m = 20;
        }
    }
}
```

Syntax error, insert "}" to complete MethodBody

Type mismatch: cannot convert from int to short

```
    short x = 5;
    short y = 10;
    short z = x * y;
```

The local variable i may not have been initialized

```
    int i;
    System.out.println(i);
```

- Shgirayot Kompilatsiya (hizer): Shgirayot Shnait "L'tifos" Beht
  Kiriat Hokok VaHifivat Root bytecode Levm, hemedar

Doggonaot:

- B'derek Kelal Kesivot Le:
  Tahbir, Te'amot Tiposim, Gederah Lefni Shimos
Compilation vs. Runtime Errors

شرعיות زمن ريئة: לא ניתן לדעת שיתפ农贸ת שגיאה במקומآ ספציפي
בזמן ההידור (קומפליציה)

דוגמאות:

```java
int a[] = new int[10];
...
a[15] = 10;
...

a = new int[20];
...
```

```java
String s = null;
System.out.println(s.length());
...
```

( Exceptions, מתכון למנגנון החירויות (Exceptions), עליז נשלד ב홈יש)
Compilation vs. Runtime Errors

**האם יש עוד סוג של טעויות?**

ן, הכי גרועות, טעויות לוגיות בתוכנית

```java
public class T {
    /** calculate x! */
    public static int factorial(int x) {
        int f = 0;
        for (int i = 2; i <= x; i++)
            f = f * i;
        return f;
    }
}
```
The Debugger

- Some programs may compile correctly, yet not produce the desirable results.
- These programs are valid and correct Java programs, yet not the programs we meant to write!
- The debugger can be used to follow the program step by step and may help detecting bugs in an already compiled program.
Debugger – Add Breakpoint

- Right click on the desired line
- “Toggle Breakpoint”
Debugger – Start Debugging

public class Test {
    public static void main (String [] args){
        System.out.println(computeFibElement(7));
    }

    public static int computeFibElement(int n) {
        if (n == 0 || n == 1)
            return 1;
        int prev = 1;
        int prevPrev = 1;
        int curr;
        for (int i = 2 ; i < n ; i++) {
            curr = prev + prevPrev;
            prevPrev = prev;
            prev = curr;
        }
        curr = prev + prevPrev;
        return curr;
    }
}
Debugger – Debug Perspective

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?
Debugger – Debugging

Current state

Back to Java perspective

Current location
Debugger – Debugging
Using the Debugger: Video Tutorial

You can find excellent video tutorials on how to use the debugger at:

http://eclipsedtutor.sourceforge.net/debugger.html*

It is recommended to watch at least the first four videos.
The link also appears on the course website in the section on the development environment.

Detailed information on how to develop tutorials and experimental projects.

* A warning is displayed on the first page of the course on the last page.