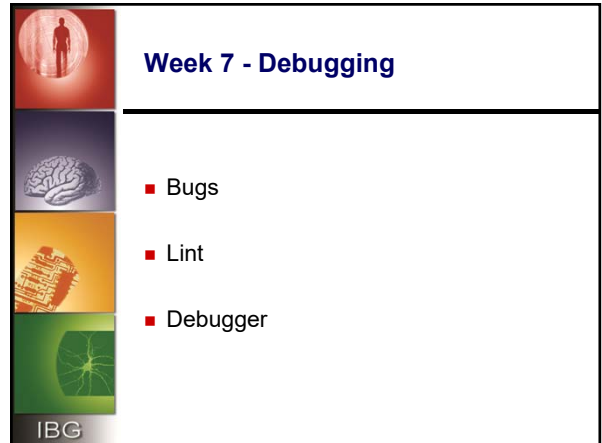


Introduction to programming
2017/18
Debugging

Izhar Bar-Gad
Room: 408 Phone: 7141 Email: izhar.bar-gad@biu.ac.il

The slide features a grid of four images: a red square with a white silhouette of a person, a green square with a white lightning bolt, a blue square with a white brain, and a yellow square with a white circuit board.

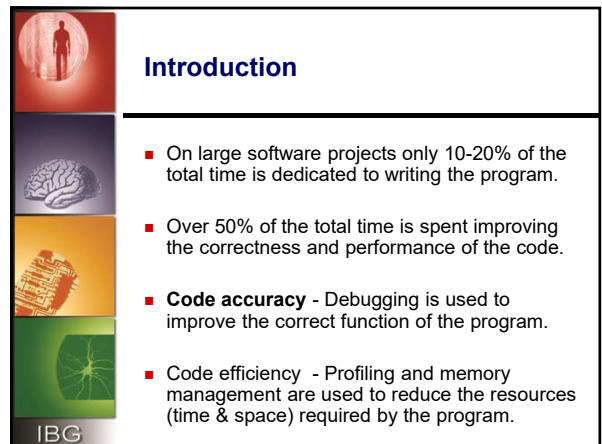


Week 7 - Debugging

- Bugs
- Lint
- Debugger

IBG

The slide features a vertical stack of four images: a red square with a white silhouette of a person, a blue square with a white brain, a yellow square with a white circuit board, and a green square with a white lightning bolt.




Introduction

- On large software projects only 10-20% of the total time is dedicated to writing the program.
- Over 50% of the total time is spent improving the correctness and performance of the code.
- **Code accuracy** - Debugging is used to improve the correct function of the program.
- **Code efficiency** - Profiling and memory management are used to reduce the resources (time & space) required by the program.




IBG

The slide features a vertical stack of four images: a red square with a white silhouette of a person, a blue square with a white brain, a yellow square with a white circuit board, and a green square with a white lightning bolt.




Debugging

- Debugging is the process of improving the accuracy of the code by removal of errors (bugs) from the program.
- Different types of errors exist:
 - Syntax errors
 - Illegal results
 - Wrong results



IBG






Errors - examples

- Syntax error – might occur only during a specific condition


```
if (b>0)
    y=1;
else
    y=1u;
end
??? y=1u;
|
Error: Unexpected MATLAB expression.
```

- Illegal results – under some runtime condition the expression is illegal

```
for i=-4:4
    g(i) = 8/i;
end
??? Subscript indices must either be real positive integers or logicals.
```






IBG







Debugging methods

- Debug prints
- Verbose mode
- Lint
- Interactive debugger







IBG



Debug prints

- Dirty method: Adding/removing, the echo variable symbol → “;”
- Slightly more elegant: Adding error message “disp”, “fprintf” or error graphics “plot”, “image” commands and commenting them out as needed.

IBG







Verbose mode

- Slightly more elegant: Add debug message and other commands (such as saving to file) when running the program in a special mode.
- Example

```
...
if (vrb)
    disp('Plotting the vector xyz');
    plot(xyz);
    save('tmpFile','xyz');
end
...
```


IBG



lint


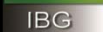
- Lint is a **programming tool** that performs the lexical and syntactic portions of the compilation with substantial additional checks.
- The name of the program was derived from the notion that it would lead to “cleaner”, more desirable programming, by picking the “lint” - little bits of fluff - out of one’s code.

IBG



mlint

- Display inconsistencies and suspicious constructs in M-files.
- MATLAB version of a popular tool which exists in many language.
- mlint – text based interface
- mlintprt – graphical interface



mlint - example

```
% Function used for demonstrating the usage of mlint
% for aiding the debugging process.

function y = ibgLintExample(x);



k = 3;

t = prod(size(g));

for i=1:100
    y(i)=i;
end

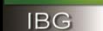
if (t & g)
    y = 5;
end


>> mlint ibgLintEx
L 4 (C 14-27): Function name 'ibgLintExample' will be known to MATLAB by its file name: 'ibgLintEx'.
L 4 (C 29): Input variable 'x' appears never to be used.
L 4 (C 31): Extra semicolon is unnecessary.
L 6 (C 1): The value assigned here to variable 'k' might never be used.
L 8 (C 5-8): NUMEL(x) is usually faster than PROD(SIZE(x)).
L 11 (C 5): Array 'y' might be grown using subscripting. Consider preallocating for speed.
L 14 (C 7): Use && instead of & as the AND operator in (scalar) conditional statements.
```






lint - notes

- mlint does not run the program.
- mlint analyses the file itself and does not check called functions.
- mlint does not catch all the bugs but rather checks for some known potential pitfalls.






Interactive debugger

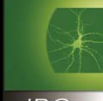




IBG

- A debugger is a computer program that is used to debug other programs.
- The debugger allows manual testing of parts of the programs and also allows searching for some conditions.
- Progressing is either manual or until a breakpoint is met.




Debugger - breakpoints






IBG

- **dbstop** – Set a break point
 - *dbstop if error* – stop when an error is encountered.
 - *dbstop in myProg at 10* - stop in file name "myProg" at the 10th line.
 - *dbstop in myProg if g==h* – stop in file name "myProg" if g and h become equal
- **dbclear** – Remove a breakpoint
 - *dbclear all* – remove all breakpoints.
 - *dbclear if error* - remove breakpoint when an error occurs.
- **dbquit** – Exit debugging mode.







Debugger - navigation



IBG

- **dbcont** - Continue execution of M-file until completion or until another breakpoint is encountered.
- **dbstep** - Execute the current line of the M-file.
- **dbstep in** - Execute the current line of the M-file and, if the line is a call to another function, step into that function.
- **dbstep out** - After stepping in, run the rest of the called function or subfunction, leave the called function, and pause.

	Debugger – function stack
	■ Functions for navigating through the call stack
	■ dbstack – List the calling stack.
	■ dbup – Go up the stack (calling function).
IBG	■ dbdown – Go down the stack (called function).