Debugging

Dr. Abdallah Mohamed

Acknowledgement: Original slides provided courtesy of Dr. Lawrence.

Key Points

1) Debugging is the act of finding and correcting errors in a system.

2) All users need to know the general debugging steps due to the complexity of computer systems.

3) A common **reason for computer errors** is our **lack of precision** in specifying instructions to the computer.

Computers are Dumb… so We Must be Precise

Computers have no knowledge or intelligence **unless they are programmed with it**.

When talking with people, we **assume knowledge** and the **ability to reason** out errors or missing details when communicating.

Computers hate imprecision and cannot handle it by default.

- Programmers often write applications to **detect** simple, common imprecise statements and fix them (but not always).

Entering Data into Forms

Data is typically **entered** into a computer **using** a **form**.

A programmer can **restrict the types and number** of symbols that can go into a form field.

- e.g. only allow numbers in a phone number field

Many errors occur when users either enter data that does not follow these restrictions, or they enter incorrect data that is accepted by the computer **because it is not properly checked**.

Question: Have you ever entered false data into a form?

Debugging: What's the Problem?

Debugging is the process of determining why **a system** does not work properly.

We perform debugging all the time in daily life, usually to fix problems with other systems and tools we interact with (cars, lights, appliances, electronics, our own bodies, etc.).

Debugging is a little different with computers and information technology because **usually it is not a component failure** that is the source of the problem. More commonly, **it is our interaction and limited understanding** of how the computer works.

Whose Problem Is It?

When we build or use an information system, we are always part of the problem!

- We give the commands and the input, so the only other possible cause is a broken system.

People ***do not*** *knowingly* make errors, but we frequently do if we do not understand how to use a system properly.

- We must be precise and know what the computer expects.

Debugging is challenging as a computer user because:

- the computer cannot debug itself
- we cannot debug it directly either because the error is internal to the computer

Debugging involves *working with* the computer to try and understand what is happening and why.

Debugging: Solving a Mystery

Debugging is very similar to solving a mystery.

To discover and solve the problem we ask questions like:

- ◆Do I need more clues?
- ◆Are my clues reliable?
- ◆What is a theory to explain the problem?
- ◆How can I test if my theory is correct?

Like solving mysteries, the only way **to get good at debugging** is **practice and gaining experience** about common problems and solutions.

The Four Key Steps in Debugging

1) Check that the error is reproducible.
- ◆Computers are deterministic. Make sure you ***know exactly how to reproduce the error***. **Identify the error!**

2) Do not jump to conclusions.
- ◆**Make sure you know what the problem is**.

3) Check all the "obvious" sources of error.
- ◆You would be surprised how often a cable is not plugged in…

4) Isolate the problem
- ◆The goal is to make good assumptions and divisions of parts that you know are working and others that need investigation.
- ◆Be careful! It is often parts (including yourself) that you assume are working that really are not.
  - ⇨Make sure assumptions are backed up by tests.

Debugging HTML Web pages

How to debug HTML web pages according to the 4 steps:

- ◆1) Reproduce errors - This is easy. Every time you reload or refresh the page, you should see the same errors.
- ◆2) Do not jump to conclusions - Although there are bugs in web browsers, it is vastly more likely that the **HTML document** contains errors. **Focus your attention there**.
- ◆3) Obvious sources of errors - One "obvious" source of errors is **non-matching open and closed tags**.
  - ⇨As you gain experience, more errors become obvious.
- ◆4) Isolate the problem - An HTML document is processed starting at the beginning, so try to fix errors at the start of the document first then work down.

Common HTML Errors

Some common HTML errors:

- ◆Open with no matching close tag
  - ⇨`<a href="test.html"> link <p> pargraph </p></html>`
- ◆Non-matching quotes
  - ⇨`<img src="myimage.gif>` (no closing quote)
  - ⇨`<img src="myimage.gif'>` (open with ", close with ')
  - ⇨`<img src=“myimage.gif”>` (HTML does not like smart quotes)
- ◆Missing attribute or incorrect attribute name.
- ◆Incorrect tag name (which may result in non-matching tags).
- ◆Incorrect file name or hyperlink address.
- ◆Forgot required tags like `<html>`, `<head>`, `<body>`.
- ◆Forget to stop escape sequence with a semi-colon
  - ⇨e.g. `&lt` (missing semi-colon should be `&lt;`)

Aside: The Cost of Debugging

When developing a computer system or application, the process of **testing and debugging is extremely costly**.

Most software requires **40% of the total time, cost, and effort** to debug and fix problems in the system.

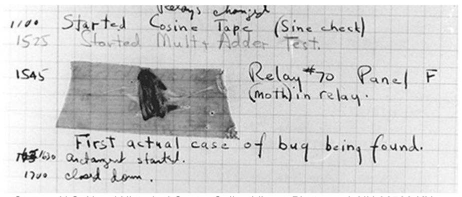
- ◆Even so, many errors go unnoticed until the system is used.

To make software development more efficient and less costly, ***software engineering*** principles and techniques are followed.

- ◆Although building software is harder than building a bridge due to its complexity, software engineers continually strive to make software development better.

Aside: The First Bug

The first "bug" in a computer system was actually a moth found in the *Harvard Mark II, an* an ***electromechanical*** computer system, in 1947 by Rear Admiral ***Grace Hopper***.



Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult + Adder Test.
1545 Relay #70 Panel F (moth) in relay.
First actual case of bug being found.
1630 antangent started.
1700 closed down.

Source: U.S. Naval Historical Center Online Library Photograph NH 96566-KN

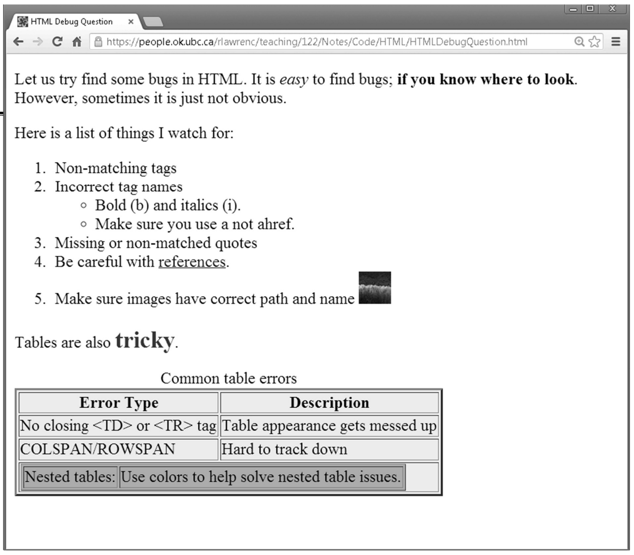
Debugging Follows the Scientific Method

Determining what a program does and **finding any errors (i.e., *testing*)** follows the scientific method.

1) **Model** – **create a hypothesis** on what the program does
2) **Predict** - for inputs not yet tried / simulated
3) **Experiment** – run the program to check your prediction
4) **Refine** – modify your hypothesis based on experimental results and repeat.

Understanding software is in many ways similar to understanding how complex real-world processes work.

HTML Debugging Question
Desired Output



https://people.ok.ubc.ca/rlawrenc/teaching/122/Notes/Code/HTML/HTMLDebugQuestion.html

Let us try find some bugs in HTML. It is *easy* to find bugs; **if you know where to look**. However, sometimes it is just not obvious.

Here is a list of things I watch for:

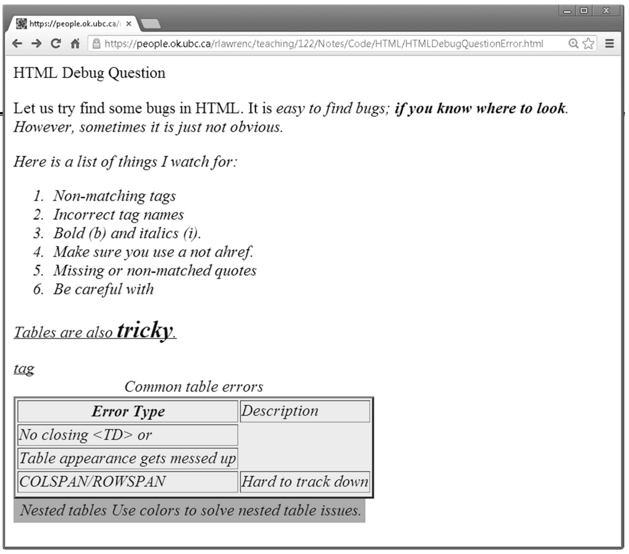
1. Non-matching tags
2. Incorrect tag names
   - Bold (b) and italics (i).
   - Make sure you use a not ahref.
3. Missing or non-matched quotes
4. Be careful with references.
5. Make sure images have correct path and name

Tables are also **tricky**.

Common table errors

Error Type	Description
No closing <TD> or <TR> tag	Table appearance gets messed up
COLSPAN/ROWSPAN	Hard to track down
Nested tables:	Use colors to help solve nested table issues.

HTML Debugging Question
Actual Output



https://people.ok.ubc.ca/rlawrenc/teaching/122/Notes/Code/HTML/HTMLDebugQuestionError.html

HTML Debug Question

Let us try find some bugs in HTML. It is *easy to find bugs; **if you know where to look**. However, sometimes it is just not obvious.*

Here is a list of things I watch for:

1. *Non-matching tags*
2. *Incorrect tag names*
3. *Bold (b) and italics (i).*
4. *Make sure you use a not ahref.*
5. *Missing or non-matched quotes*
6. *Be careful with*

*Tables are also **tricky**.*

tag

Common table errors

Error Type	*Description*
No closing <TD> or	
Table appearance gets messed up	
COLSPAN/ROWSPAN	*Hard to track down*

Nested tables Use colors to solve nested table issues.

HTML Debugging Question
HTML Document

```
<html>
<head><tile>HTML Debug Question</title></head>  <body>
<p>Let us try find some bugs in HTML.  It is <i>easy to find </i>
bugs; <b>if you know where to look</b>.  However, sometimes it is
just not obvious.</p>
<p>…</p>
Here is a list of things I watch for:
<ol>
<li>Non-matching tags</li>
<li>Incorrect tag names <li>
<ul>    <li>Bold (B) and italics (I).</li>
        <li>Make sure you use A not AHREF.</li>
    </ul>
<li>Missing or non-matched quotes</li>
<li>Be careful with <a href="HelloWorld.html'>references</a>.
</li>
<li>Make sure images have correct path and name
        <img src="winter.jpeg'' width="32" height=32></li>
</ol>
```

HTML Debugging Question
HTML Document (2)

```
<p>Tables are also <b style="font-size:150%;
color:redd">tricky</b>.</p>

<table border=2 style="background-color:yellow">
 <caption>Common table errors</caption>
 <tr><th>Error Type</th><td>Description</td></tr>
 <tr><td>No closing &ltTD&gt; or <TR> tag</td>
               <td>Table appearance gets messed up</td></tr>
 <tr><td>COLSPAN/ROWSPAN</td><td>Hard to track down</td></tr>
 <tr><table style="background-color:orange">
     <tr><td colspan=2><td>Nested tables</td>
     <td>Use colors to help solve nested table issues.</td></tr>
     </table></td></tr></table>
</table>

</body>
</html>
```

Conclusion

Debugging is a systematic approach to discover and fix errors in a system.

- ◆Debugging a computer system requires working with the computer to diagnose the problem with the realization that we are often the cause of the problem.

The four key steps of debugging are:

- ◆1) Check that the error is reproducible.
- ◆2) Make sure you know what the problem is.
- ◆3) Check all the "obvious" sources of error.
- ◆4) Isolate the problem

As users, we can resolve many errors with a little practice, experience, and patience without requiring help from IT service technicians.

Objectives

- ◆Give some examples of imprecise communication.
- ◆Explain why precision is important for a computer.
- ◆Define: debugging
- ◆List and explain the 4 key steps of debugging.
- ◆List (and remember) some common HTML errors.

☆

Be prepared to debug HTML documents both on the computer and on paper.