Chapter 6

An Introduction to Debugging
Learning Objectives

• Explain how ordinary precision differs from computing precision
• Describe the six-step strategy for debugging
  – Explain the purpose of each step
  – Give an example of each step
• Apply the six-step strategy for debugging the HTML code for a Web page
• Learn how to approach debugging when you don’t understand the system
• Appreciate the problems of making perfectly reliable computing systems
Precision: The High Standards of Computing

- Computers do exactly what they are told
- They know what we said, not what we meant
- In this chapter:
  - The importance of saying precisely what we mean
  - The process of figuring out what we said that we didn’t mean
Be Accurate

• Be aware, of some possible mistakes
  – Recognizing mistaken substitutions: l for 1, O for 0, \ for /, and so forth
  – Knowing that certain fonts can be very confusing: Corbel zero (o) and oh (o)
  – Respecting upper- and lowercase in pathnames (www.ex.org/AllMine.html is not the same as www.ex.org/allmine.html)
  – Respecting upper- and lowercase in passwords
Be Observant

• A principle of computing is that you should expect feedback when interacting with software
• We rely on this feedback (busy icons, etc.)
• By paying attention to feedback, we can catch errors as we make them
What’s the Problem?

- Debugging is a process by which one figures out why something isn’t working properly.
- Debugging relies mostly on logical reasoning and is learned through experience.
• Debugging is troubleshooting
• Humans do it all the time (why didn’t the car start? why didn’t the alarm go off?)
• Faults and failures in everyday life usually involve devices that are working systems with a broken or worn-out part
• The system is properly designed and constructed, but some part failed
Debugging in IT

- Debugging in computation is different
- We may have entered wrong data or wrong configuration information into a working system.
  - When it’s corrected, the system works.
- Or we might have a logical design error
  - we don’t get what we think we should get
- Always begin by assuming the system is correct and working
Whose Problem Is It?

- When debugging a computing system, *we* are almost always part of the problem.
- *We* command the computer to do tasks and *we* input the information.
- Two of the three possible problems with the system involve us:
  1. *wrong data*
  2. *wrong command*
  3. broken system
Whose Problem Is It?

• We don’t knowingly make errors…therefore we must be right! And the computer is at fault!
• Sometimes it is: both hardware and software errors do happen
• Human errors occur more often
Using the Computer to Debug

• Not only is the computer unable to debug itself, we can’t debug it directly
• Workaround: Bypassing an error and continuing to use the system
More About Debugging

• Debugging is solving a mystery
• By asking questions ("Do I need more clues?"; "Are my clues reliable?) we focus and discover a solution faster
Steps to Debugging

1. The first step in debugging is to check that the error is reproducible
   – Computers are deterministic -- they do exactly the same thing every time if given the same input
   – Start by trying to reproduce the problem
Steps to Debugging

2. *The next step is to be sure that you know exactly what the problem is*
   
   – the computer may perform a sequence of operations *after* an error
   
   – These operations must be eliminated first as the focal point of the debugging
   
   – Determining the exact problem is critical
Steps to Debugging

3. A standard next step is to check all of the “obvious” error sources
   – What kinds of errors are obvious depends on the problem
   – Checking inputs, connections, links, etc.
Steps to Debugging

4. Isolate the problem by dividing the operation into those parts that are working and those that are not
   – Theorize about where the problem is located
   – Gathering more information if needed
   – Limit the number of untested assumption
   – Eliminate as many possibilities as you can
Debugging Recap Guidelines

• Make sure that you can reproduce the error
• Determine the problem exactly
• Eliminate the “obvious” causes
• Divide the process, separating the parts that work from the part that doesn’t
• When you reach a dead end, reassess your information, asking where you may be making wrong assumptions or conclusions; then step through the process again
• As you work through the process from start to finish, make predictions about what should happen and verify that your predictions are fulfilled
A Debugging Case Study

• To illustrate the debugging principles in action, imagine a simple page in HTML:

• The goal is to get a page that looks like this:

![Figure 6.3 The target page displayed (correctly) with Firefox 25.0.](image)
Look Closely At the Page

Working Page

Not Working Page. Firefox

Figure 6.3 The target page displayed (correctly) with Firefox 25.0.
Look Closely At the Page

Working Page

It sounds bold to claim that Jackie Joyner-Kersee is the absolute best female athlete, but consider this: She competed in the heptathlon, a track and field event that combines scores from seven different sports. She won two Olympic gold medals in heptathlon (and a silver), and still holds the world record for greatest number of points over scored: 7,291.

How good was she? First, she competed in heptathlon, meaning she was Olympic caliber in 100m hurdles, 200m, 800m, high jump, long jump, javelin and shot put. Also, she won Olympic gold in long jump and two bronzes. Add to that two World Championships gold in heptathlon and two bronze in long jump, and a long jump gold in the Pan American Games. She also played starring forward all four years of college at UCLA in basketball.

No One Better. But probably the most impressive fact about her abilities is that only two other women have ever been able to score more than 7000 points in heptathlon, Carolina Kluit (7032) of Sweden and Larisa Turchinskaya (7007) of the Soviet Union. For comparison, the table at right lists the seven sports of heptathlon, Jackie's personal best, and the performance needed in each of the seven sports to earn 1000 points. (The scoring in heptathlon is bizarre.)

Inspiration Jackie Joyner-Kersee has said in her autobiography A Kind of Grace that as a young girl she was inspired to be a versatile athlete by a movie about Babe Didrikson Zaharias, who was a track star, basketball player and pro golfer, and ironically, considered the All-Time Best Female Athlete before Jackie.

Not Working Page..Safari

Figure 6.3 The target page displayed (correctly) with Firefox 25.0.

Figure 6.5 The buggy page displayed with Safari 6.0.
Look Closely At the Page

Working Page

Jackie Joyner-Kersee -- All-Time Best Female Athlete

It sounds bold to claim that Jackie Joyner-Kersee is the absolute best female athlete, but consider this: She competed in the heptathlon, a track and field event that combines scores from seven different sports. She won two Olympic gold medals in heptathlon (and a silver), and still holds the world record for greatest number of points ever scored: 7,291.

How good was she? First, she competed in heptathlon, meaning she was Olympic caliber in 100m hurdles, 200m, 800m, high jump, long jump, javelin and shot put. Also, she won Olympic gold in long jump and two bronze. Add to that two World Championship golds in heptathlon and two silver. In long jump, and a long jump gold in the Pan American Games. She also played softball for all years of college at UCLA in baseball.

No One Better. But probably the most impressive fact about her abilities is that only two other women have ever been able to score more than 7000 points in heptathlon. Carolina Kluft (7032) of Sweden and Larisa Turchinskaya (7007) of the Soviet Union. For comparison, the table at right lists the seven sports of heptathlon. Jackie’s 7291-year (1988), her personal best, and the performance needed in each of the seven sports to earn 1000 points. (The scoring in heptathlon is bizarre.)

Inspiration Jackie Joyner-Kersee has said in her autobiography A Kind of Grace that as a young girl she was inspired to be a versatile athlete by a movie about Babe Didrikson Zaharias, who was a track star, basketball player and golfer, and ironically, considered the All-Time Best Female Athlete before Jackie.

Figure 6.3 The target page displayed (correctly) with Firefox 25.0.

Not Working Page..Chrome

Jackie Joyner-Kersee -- All-Time Best Female Athlete

It sounds bold to claim that Jackie Joyner-Kersee is the absolute best female athlete, but consider this: She competed in the heptathlon, a track and field event that combines scores from seven different sports. She won two Olympic gold medals in heptathlon (and a silver), and still holds the world record for greatest number of points ever scored: 7,291.

How good was she? First, she competed in heptathlon, meaning she was Olympic caliber in 100m hurdles, 200m, 800m, high jump, long jump, javelin and shot put. Also, she won Olympic gold in long jump and two bronze. Add to that two World Championship golds in heptathlon and two silver. In long jump, and a long jump gold in the Pan American Games. She also played softball for all years of college at UCLA in baseball.

No One Better. But probably the most impressive fact about her abilities is that only two other women have ever been able to score more than 7000 points in heptathlon. Carolina Kluft (7032) of Sweden and Larisa Turchinskaya (7007) of the Soviet Union. For comparison, the table at right lists the seven sports of heptathlon. Jackie’s 7291-year (1988), her personal best, and the performance needed in each of the seven sports to earn 1000 points. (The scoring in heptathlon is bizarre.)

Inspiration Jackie Joyner-Kersee has said in her autobiography A Kind of Grace that as a young girl she was inspired to be a versatile athlete by a movie about Babe Didrikson Zaharias, who was a track star, basketball player and golfer, and ironically, considered the All-Time Best Female Athlete before Jackie.

Figure 6.6 The buggy page displayed with Chrome 30.0.
Look Closely At the Page

Working Page

Jackie Joyner-Kersee -- All-Time Best Female Athlete

It sounds bold to claim that Jackie Joyner-Kersee is the absolute best female athlete, but consider this: she competed in the heptathlon, a track and field event that combines scores from seven different sports. She won two Olympic gold medals in heptathlon (and a silver), and still holds the world record for greatest number of points over seven events: 7,291.

How good was she? First, she competed in heptathlon, meaning she was Olympic caliber in 100m hurdles, 200m, 800m, high jump, long jump, javelin and shot put. Also, she won Olympic gold in long jump and two bronze. Add to that two World Championship golds in heptathlon and two more in long jump, and a long jump gold in the Pan American Games. She also played starring role all four years of college at UCLA in basketball.

No One Better. But probably the most impressive fact about her abilities is that only two other women have ever been able to score more than 7000 points in heptathlon, Carolina Klüft (7032) of Sweden and Larisa Turchinskaya (7007) of the Soviet Union. For comparison, the table at right lists the seven sports of heptathlon. JJK’s 7291-year (1988), her personal best, and the performance needed in each of the seven sports to earn 1000 points. (The scoring in heptathlon is bizarre.)

Inspiration: Jackie Joyner-Kersee has said in her autobiography A Kind of Grace that as a young girl she was inspired to be a versatile athlete by a movie about Babe Didrikson Zaharias, who was a track star, basketball player and pro golfer, and ironically, considered the All-Time Best Female Athlete before JJK.

Figure 6.3 The target page displayed (correctly) with Firefox 25.0.

Not Working Page..IE

Jackie Joyner-Kersee -- All-Time Best Female Athlete

It sounds bold to claim that Jackie Joyner-Kersee is the absolute best female athlete, but consider this: she competed in the heptathlon, a track and field event that combines scores from seven different sports. She won two Olympic gold medals in heptathlon (and a silver), and still holds the world record for greatest number of points over seven events: 7,291.

How good was she? First, she competed in heptathlon, meaning she was Olympic caliber in 100m hurdles, 200m, 800m, high jump, long jump, javelin and shot put. Also, she won Olympic gold in long jump and two bronze. Add to that two World Championship golds in heptathlon and two more in long jump, and a long jump gold in the Pan American Games. She also played starring role all four years of college at UCLA in basketball.

No One Better. But probably the most impressive fact about her abilities is that only two other women have ever been able to score more than 7000 points in heptathlon, Carolina Klüft (7032) of Sweden and Larisa Turchinskaya (7007) of the Soviet Union. For comparison, the table at right lists the seven sports of heptathlon. JJK’s 7291-year (1988), her personal best, and the performance needed in each of the seven sports to earn 1000 points. (The scoring in heptathlon is bizarre.)

Inspiration: Jackie Joyner-Kersee has said in her autobiography A Kind of Grace that as a young girl she was inspired to be a versatile athlete by a movie about Babe Didrikson Zaharias, who was a track star, basketball player and pro golfer, and ironically, considered the All-Time Best Female Athlete before JJK.

Figure 6.7 The buggy page displayed with Internet Explorer 10.0.
Look Closely At the Page

• The best way to get started, is to study the output closely to see where the errors are.

• The goal is to notice features that are **right** and features that are **wrong**
  – Note that the four browsers display the buggy page differently
  – All browsers should show the page exactly the same
  – It is sometimes possible to find a bug by comparing how different browsers show it
Jackie Joyner-Kersee --- All-Time Best Female Athlete

It sounds bold to claim that Jackie Joyner-Kersee is the absolute best female athlete, but consider this: She competed in the heptathlon, a track and field event that combines scores from seven different sports. She won two Olympic gold medals in heptathlon (and a silver), and still holds the world record for greatest number of points ever scored: 7,291.

How good was she? First, she competed in heptathlon, meaning she was Olympic caliber in 100m hurdles, 200m, 800m, high jump, long jump, javelin and shot put. Also, she won Olympic gold in long jump and two-bronze in the Pan American Games. She also played starring role in UCLA basketball.

But probably the most impressive fact about her abilities is that only two other women have ever been able to score more than 7000 points in heptathlon, Carolina Klüft (7032) of Sweden and Larissa Turchinskaya (7007) of the Soviet Union. For comparison, the table at right lists the seven sports of heptathlon, JK’s 1988, her personal best, and the performance needed in each of the seven sports to earn 1000 points. (The is bizarre.)

Jackie Joyner-Kersee has said in her autobiography that as a young girl she was inspired to be a versatile athlete by a movie about Babe Didrikson Zaharias, who was a track star, basketball player and pro golfer, and ironically, considered the All-Time Best Female Athlete before JK.

Figure 6.8 The buggy HTML code.
Reproduce the Error

• As we begin debugging the HTML:
  – Recall that the first step is to reproduce the error
  – Close the browser and reopen the file
  – There is definitely a problem with our HTML!
Determine the Exact Problem

• The next step is to determine the problem exactly
  – Look at the displayed page
  – When there are multiple bugs, pick one to concentrate on
  – We start with the missing book cover image
Eliminate the Obvious

• Once the problem is known:
  – Look for the “obvious” causes
  – Eliminate them

• What’s the most obvious problem with a missing image?
  – The file is not where it should be, so the browser can’t find it
  – Check to see that the image is in the *pic* folder, which it is
Eliminate the Obvious

- The next most obvious error is misspelling the file name:
  - Check the HTML
  - The path is messed up: It doesn’t mention the pic folder
- After fixing the tag, the image is there!
- It was an obvious mistake, so checking the “obvious” problems solve it
- One error down!
Focusing the Search

• We return to the “determine the problem step” and continue with the next bug.
  – Determine the Exact Problem
    • We choose the no styling problem
  – Eliminate the Obvious
    • Tags seem balanced
    • Open the web console to show error messages from Firefox
Focusing the Search

• Style problem, continued
  – Fix errors that make sense, which is only some
  – Divide Up the Process
    • Separate those parts of the system that work from the part that does not
    • This is not always possible to do perfectly
The “Divide Up the Process” Process

• Step through the style section one element at a time
• Do the following for each element:
  – Delete the entire element
  – Save the file
  – Refresh the display
  – Check to see that the only thing different is the styling of the removed element
  – Undo the deletion to restore the file to its original form
The “*Divide Up the Process*” Process

- If removing an element restores other formatting, then we have found the element that is stopping the styling
  - the th style is the culprit
- We can fix it!
  - Separate the part that is broken from the parts that work (sound familiar?)
The “Divide Up the Process” Process

• To figure out what is wrong with an element in css:
  – Repeat the process we just went through, eliminating parts of this line to see which one is the problem
  – Another approach might be to re-enter the text
    • occasionally files get odd
    • unprintable characters can be hard to locate
    • retyping requires that we consider each part of the styling again, carefully
The “Divide Up the Process” Process

- Turns out the opening brace was really a comma
- Somehow, this made the browser miss the closing brace
- Similar procedures fix the caption location and text coloring errors
Debugging the Page: A Postmortem

• We debugged the mistakes in the page by applying debugging guidelines:
  1. Checked that the book cover file was in the `pic` folder—it was.
  2. Checked the spelling of the file, but found that the path left out the `pic` part. *Fixed.*
  3. Checked that the end tags were all present—they were.
  4. Clicked web console to get diagnostics
Debugging the Page:
A Postmortem

• We debugged the mistakes in the page by applying debugging guidelines:
  5. Checked padding, and noticed that the unit had not been set. Fixed.
  6. Returned to the Error Console to find two mysterious diagnostics.
  7. Checked that the closing braces were all present—they were.
Debugging the Page: A Postmortem

• We debugged the mistakes in the page by applying debugging guidelines:
  8. Separated working from failing code by a process of delete, check, undo—found *th* had stopped styling.
  9. Checked the *th* element, and eventually spotted the opening parenthesis. *Fixed.*
  10. Looked at the Error Console again to find new diagnostics.
Debugging the Page: A Postmortem

- We debugged the mistakes in the page by applying debugging guidelines:
  
  11. Looked up “caption-align” to find it is “caption-side.” *Fixed.*
  
  12. Checked the wrongly colored text, and spotted a typo in the end tag. *Fixed.*
  
  13. Checked the caption’s small font specification, and spotted a colon typo. *Fixed.*
Debugging the Page: A Postmortem

• An additional common debugging feature: “Correcting” text that isn’t actually wrong because of a wrong conjecture
• When we’re lucky, we end up changing the program text from correct to correct
• When we’re unlucky, we change it from correct to incorrect
  – This means “backing out” and restoring the original
Debugging the Page: A Postmortem

• Hiding Other Errors
  – Notice that the errors found were different from the errors we thought we had originally
  – This is very common in debugging:
    • Programmers never say that they are “down to the very last bug” . . . that bug could be hiding one more!
Debugging the Page: A Postmortem

• Asking the System to Help
  – The most effective technique was to use the browser’s Error Console feature
  – It would have been more effective if the we had better understood the results
  – The Page Source shows color- and font-coded HTML source that tells us how the browser interprets the page
Debugging the Page: A Postmortem

- Asking the System to Help
  - One of the most powerful debugging techniques is to find ways for the computer to tell us the meaning of the information it stores or the effects of the commands it executes.
  - Having the computer say how it’s interpreting our instructions can separate the case in which we give the right command—but mess up expressing it—and giving the wrong command.
No Printer Output...

• You try to print a document and nothing happens...

• The printing problem is solved just like the earlier problems were solved:
  – Reproduce the error,
  – Understand the problem, and
  – Check the obvious causes
No Printer Output…

• These steps include:
  – checking the printer’s control panel, the paper, the cartridges, the cable connections, the file to be printed, the installation of the printer driver,
  – whether others can print if this is a shared printer, and
  – whether you can print a different document
No Printer Output…

• Take the next step in the debugging strategy: Try to isolate the problem
  – Because you have printed before, you know your computer is configured correctly.
  – Try to print a simple document, but it’s the same story
  – Is the computer even trying to send the file to the printer?
  – Unplug the printer from the computer and try again to print.
No Plug, No Complaint

• Isolating the problem
  – You expect that when printing with the printer unplugged, the computer will complain
  – But it does not, so you suppose the computer is not even trying to send the files
  – Locate the printer driver’s printing monitor to see it has files waiting to be sent
The Print Queue

• The print queue for your machine:
  – A place where printing tasks wait before being processed
  – You find it was somehow set to hold print jobs instead of print them
  – The best approach is to cancel or trash all of the jobs in the queue, and restart the queue
  – Configure the printer so that it prints your files immediately rather than queuing them
  – Did you remember to plug the printer back in?
Ensuring Software Reliability

• Software contains bugs, and crashes are frustratingly frequent
• Most errors are just an annoyance
• What about computers that control life-support systems, medical apparatus, airplanes, nuclear power plants, weapons systems, etc?
• Errors in these systems are potentially much more serious
Hardware Failures

• Hardware failures can be resolved using techniques such as redundancy…multiple computers performing computations of a safety-critical system

• Another technique is dubbed burn in
  – Most errors show up after few hours of operation
  – A computer that has a record of successful operation is likely to continue to operate successfully
Software Failures

- Software is amazingly complex
- Number of possible configurations that a typical program can define grows exponentially
- All these states, known as reachable configurations, cannot be examined for correctness
- How can we be sure programs work correctly?
Software Failures

• Programmers begin with a specification or a precise description of:
  – the input,
  – how the system should behave, and
  – how the output should be produced

• The specification doesn’t say how the behavior is to be achieved, just what it should be
Software Failures

• Using various design methods, programmers produce the program and test it with sample inputs.
• Outputs can be checked against the specification.
• If they do not match, there is a bug and the program must be fixed.
• A program is said to be correct if its behavior exactly matches its specification.
Two Serious Problems

• We cannot prove that a specification is correct
• We cannot establish program correctness by testing
• Therefore, we cannot know a program is correct
  – Even if it is
The Challenge

• What about the fact that we can’t prove that software is correct?
  – Accept that software may contain bugs
  – Poorly tested software is simply unprofessional

• Be cautious and informed users and take our business to those who produce the best product
Fail-Safe and Fail-Safe Software

• “Safe software” changes the focus from worry about program correctness to concern about the consequences
• Testing gives confidence that software works “under normal circumstances”
• It is difficult to test software under unusual circumstances
Fail-Soft and Fail-Safe Software

• There are two design strategies: fail-soft and fail-safe
  – Fail-soft means that the program continues to operate, providing a possibly degraded level of functionality
  – Fail-safe means that the system stops functioning to avoid causing harm
• The strategy is to continue to operate as long as service is safely provided
• Using software to control potentially dangerous systems means taking a risk
Community Debugging

• There are hundreds of chat sites where people post their problems, and other people offer help
• If your problem has surfaced before, then a Web search should locate a place where the solution is discussed
• Asking the community is a good tactic to keep in mind
Summary

• We learned the following:
  – What debugging is and why we need to know how to do it
  – Basic debugging strategy, including the whys and hows of debugging
  – To debug a Web page, using the Error Console of the document that shows how the computer interprets the HTML
Summary

• We learned the following:
  – How to analyze our debugging performance, noting that debugging involves both correct and incorrect conjectures
  – That it’s possible to debug a sophisticated system like a computer printer with little more than a vague idea of how it works, by using our standard debugging strategy applied with common sense and courage
Summary

• We learned the following:
  – That it is practically impossible to have bug-free software
  • This doesn’t mean that we must quit using computers or accept bugs, but we must watch for unusual behavior that might indicate bugs and take precautions to limit the harm that they can cause.