Teacher Notes

This lesson focuses on documentation specific to software. Since students are not developing any software in this unit you may prefer to delay the presentation of the material in this lesson.

You will need to collect samples of external documentation that is not software related--directions to build; manual for TV, etc.

Activity 1: you will need to make a program available to students. The program should be written in the language students used last year. It should contain no documentation and be demanding enough, they will need to struggle a bit to discover what is going on. Have the students do a paper and pencil code walkthrough. From this program, students hopefully, will see the need for internal documentation. A possible Java program is included in the lesson.

Answer Key

1. User entry input error checking.
2. 11
3. all true
4. first two are false, all others remain at true
5. to control the looping to fill in multiples with false, the square root is used to set the number of time the iteration needs to occur
6. Complete the table to track the array changes caused by lines 19 – 24.

Outside loop Iteration	Value of i	primeValue[i]	Inside loop iteration	Value of j	primeValue[j]
1	0	false	No iteration		
2	1	False	No iteration		
3	2	True	1	4	False
			2	6	False
			3	8	False
4	3	True	1	6	False
			2	9	False
5	4	True	1	8	false

7. false, false, true, true, false, true, false, true, false, false, false
8. move backwards through the loop to find the location of the first true or largest prime value

Activity 2: Webquest in which the students can take on the role of project manager to investigate and develop rules of style for internal documentation. Check links. My personal favourite is 32 bits of Style. Webquest includes suggested rubric for assessment if needed.
http://www.gecdsb.on.ca/d&g/cswebquest/ Note: just recently, Doug Peterson indicated that he will be updating this webquest for September, 2008 to meet the latest expectations for ICS4U.

Unit: Project Management ICS 4U
Topic: Documentation
Time: 2 classes

Specific Expectations

B1.3 produce the software according to specifications (i.e., code, test, deploy), and create user documentation and training materials;

Content/Student Notes

Day 1

- Internal Documentation
  - Communication between developers
  - Program maintenance
  - Decrease debugging time

Day 2

- Software External Documentation
  - User level documentation
    - Brief description of project analysis including any assumptions by team.
    - Brief description of important code features and capabilities.
    - Brief description of mechanics of running the program. For example, describe how data is entered.
    - Tutorial for large and/or complex programs. Including screen captures.
    - Interactive programs need details on commands and/or GUI interface.
    - System requirements.
    - Brief description of input/output files.
    - List of error messages that might be generated.
    - Quick guide where appropriate.
  - Developer Documentation
    - Ongoing throughout development. Keeps other developers on same page.

Teaching Strategies

- Students modify an undocumented program, or answer questions about what is going on. See teacher notes.
- Students complete Webquest.
- Note: time may be needed on day 2 to finish the webquest.

- As a class, develop a style sheet for software internal documentation. Distribute electronically.
- Give students a set of external documents, not software related. For example, directions to build a bookcase; manual for appliance. In groups, have the students make a list of common features.
- Discuss common features of external documents.
- Lecture—Software External Documentation. Students must prepare external documentation for all projects. Note: developer documentation may be maintained in project management chart.

Activity

- Complete webquest. http://www.gecdsb.on.ca/d&g/cswebquest/ .
- Complete a code walkthrough on the program provided.
-

Resources

- http://www.gecdsb.on.ca/d&g/cswebquest/

Assessment

- √Observation of task completion
- √Participation in brainstorming session concerning style.

Visual Aids/Equipment

- Handout: Sample program and code walkthrough chart.

Unit: **Project Management**
Activity: **Code Walkthrough**

```
1    // The "ExampleOne" class.
2    public class ExampleOne
3    {
4        public static void main (String[] args)
5        {
6            int number = 10;
7            try
8            {
9                number = Integer.parseInt (args [0]);
10           }
11           catch (Exception e)
12           {
13           }
14           boolean[ ] primeValue = new boolean [number + 1];
15           for (int i = 0 ; i <= number ; i++)
16               primeValue [i] = true;
17           primeValue [0] = primeValue [1] = false;
18           int n = (int) Math.ceil (Math.sqrt (number));
19           for (int i = 0 ; i <= n ; i++)
20           {
21               if (primeValue [i])
22                   for (int j = 2 * i ; j <= number ; j = j + i)
23                       primeValue [j] = false;
24           }
25           int largePrime;
26           for (largePrime = number ; !primeValue [largePrime] ; largePrime--)
27               ;
28           System.out.println ("The largest prime number less than or equal to " +
     number + " is " + largePrime);
29       }
30   } // main method
31   // ExampleOne class
```

Analyse the program, ExampleOne, to answer the following questions.

1. What is the purpose of lines 7 – 13?

2. How large is the primeValue array?

3. What are the values in the array at the start of line 17?

4. What are the values in the array after line 17?

5. What is the value and purpose of n?

6. Complete the table to track the array changes caused by lines 19 – 24.

Outside loop Iteration	Value of i	primeValue[i]	Inside loop iteration	Value of j	primeValue[j]
1	0	false	No iteration		

7. What are the values in the array after line 24?

8. What is the purpose of lines 26 and 27?

9. Rewrite the program to include internal documentation.