Visualizing Code Patterns in Novice Programmers

Jeff Bulmer, Angie Pinchbeck, Dr. Bowen Hui
University of British Columbia Okanagan

Bad Code

```
public void main(String args[]){
Scanner scan = new Scanner(System.in);
sysout("Please enter a number");
int num, temp = scan.nextInt();
int reverse = 0;
while(temp > 0){
reverse = reverse*10+temp%10;
temp/10;
}
int temp2 = reverse;
if(temp = temp2)
sysout("Is " + num + " a palindrome?\ntrue);
if(temp != temp2)
sysout("Is " +num + " a palindrome?\nfalse");
```

Bad Code

Poorly indented (makes longer code difficult to debug)

```
public void main(String args[]){
Scanner scan = new Scanner(System.in);
sysout("Please enter a number");
int num, temp = scan.nextInt();
int reverse = 0;
while(temp > 0){
reverse = reverse*10+temp%10;
temp/10;
}
int temp2 = reverse;
if(temp = temp2)
sysout("Is " + num + " a palindrome?\ntrue);
if(temp != temp2)
sysout("Is " +num + " a palindrome?\nfalse");
```

Not a valid method

Assigns a variable, doesn't check equality

Good Code

```
public void main(String args[]){
    Scanner scan = new Scanner(System.in);
    System.out.println("Please enter a number: ");
    int num = scan.nextInt();
    int temp = num;
    int reverse = 0;
    while(temp > 0){
        reverse = reverse*10+temp%10;
        temp = temp/10;
    }
    System.out.println("Is " + num + " a palindrome?");
    System.out.println(num == reverse);
    scan.close();
}
```

Good Code

Properly indented

```
public void main(String args[]){
    Scanner scan = new Scanner(System.in);
    System.out.println("Please enter a number: ");
    int num = scan.nextInt();
    int temp = num;
    int reverse = 0;
    while(temp > 0){
        reverse = reverse*10+temp%10;
        temp = temp/10;
    }
    System.out.println("Is " + num + " a palindrome?");
    System.out.println(num == reverse);
    scan.close();
}
```

Only uses 3 variables

Frees up memory

Motivation

- Prevent students from adopting bad coding practices

Motivation

- Prevent students from adopting bad coding practices

1. Find bad practices (“bad habits”)

Motivation

- Prevent students from adopting bad coding practices

1. Find bad practices (“bad habits”)

2. Address bad habits in real time

Defining Bad Habits

- What is a “bad habit”?
  - “You know it when you see it”
  - No consensus, outside of lists of common errors
  - Clearly, errors are bad habits, but what else?

Types of Bad Habits Considered

- Unclosed Scanners
- Brackets and Quotes Miscount
- Brackets and Quotes Mismatch
- Misplaced Semicolon
- Comparison vs. Assignment
- Misaligned Whitespace

Types of Bad Habits Considered

- Unclosed Scanner
  - Good

```
Scanner scan = new Scanner(System.in);
...
scan.close();
```

  - Bad

```
Scanner scan = new Scanner(System.in);
```

Types of Bad Habits Considered

- Brackets and Quotes Miscount

```
while(x < 10) {
```

- Brackets and Quotes Mismatch

```
while(x < 10) {()
```

Data Collection

- Short confidence survey
- 10 Question Coding Quiz for COSC 121 students
  - Collect answers to search for habits
  - Collect keycodes to reconstruct answers

Data Collection

- 10 Question Coding Quiz for COSC 121 students
  - Most students did not complete all 10 questions
  - All 77 students completed 5 questions

    Only 13 completed all 10

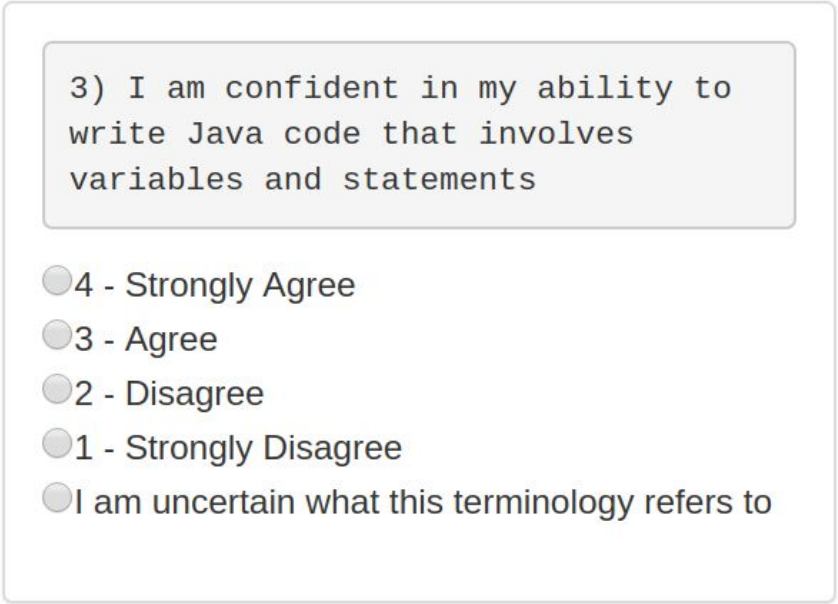
Confidence Survey

- Self-report confidence

  in 11 topics

- 4 Point Likert Scale

  - Force response



Confidence Survey

- 11 Topics:
  - Variables and Statements
  - Java Classes
  - User Input and Output
  - Characters and Strings
  - Conditionals
  - Basic Loops
  - Nested Loops
  - Methods
  - One-dimensional Arrays
  - Objects
  - Classes

Coding Survey

- 10 Questions from 11 topics
- Textbox allows tabs, does not catch errors

```
1) Write a Java program to
determine if an integer entered by
the user is a palindrome. For
example, "121" and "4334" are
palindromes, but "42" and "3355"
are not. Do not change the integer
to a String in any part of your
solution. Hint: you may want to use
% to extract a digit from the
number.
```

Results: Frequency of Bad Habits

	Frequency
Unclosed Scanners	293
Brackets and Quotes Miscounts	205
Brackets and Quotes Mismatches	44
Misplaced Semi-Colons	11
Comparison vs. Assignment	81
Misaligned Whitespace	40

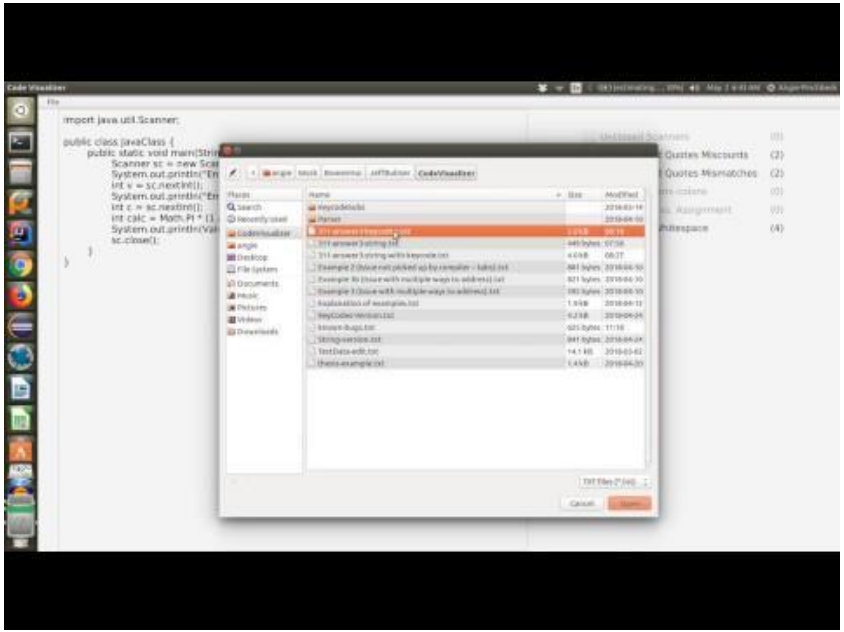
Addressing Bad Habits in Real Time

- Code Visualizer to highlight habits within each line

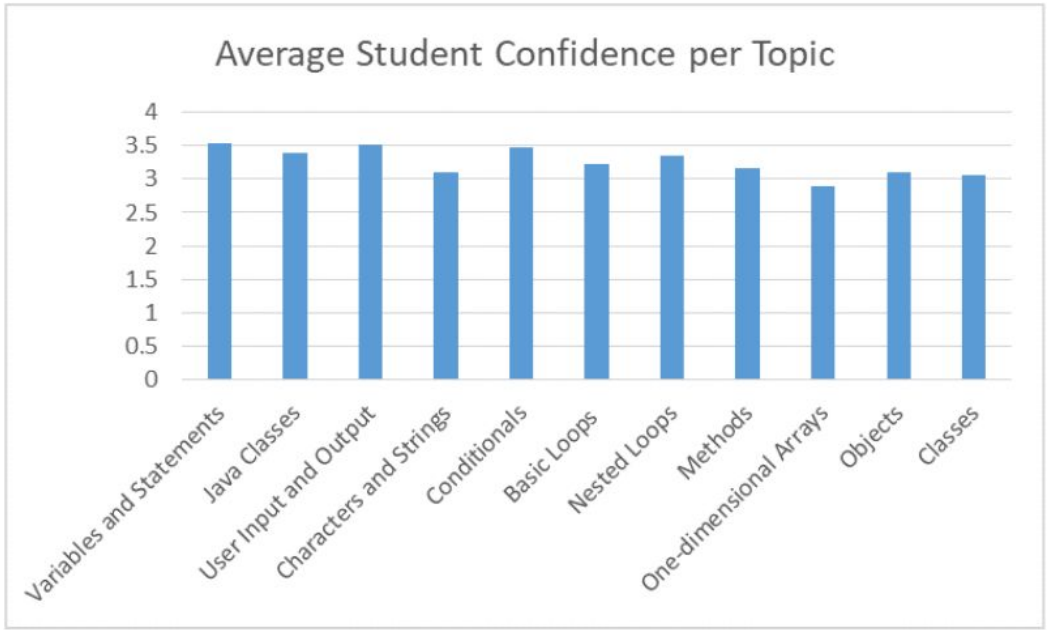
Addressing Bad Habits in Real Time

- Code Visualizer to highlight habits within each line
  - Visual component largely built by Angie Pinchbeck
- Inspired by snapshot analysis
  - Reconstruct student code
  - Look for habits at every snapshot

Visualizer Demo



Results: Confidence vs. Number of Habits



Results: Confidence vs. Number of Habits

	Q1-5	Q1-10
Unclosed Scanners	-0.061	-0.395
Brackets and Quotes Miscounts	-0.117	-0.231
Brackets and Quotes Mismatches	0.114	0.095
Misplaced Semi-Colons	0.056	-0.060
Comparison vs. Assignment	-0.258	-0.130
Misaligned Whitespace	-0.086	-0.403

Results: Lines vs. Number of Habits

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Min.	7	1	5	3	9	5	9	1	16	24
Avg.	23	18	12	14	34	9	20	19	30	39
Max.	74	74	22	33	76	16	30	32	50	49

Results: Lines vs. Number of Habits

- $r = -0.177$ for Q1-5 and $r = -0.462$ for Q1-10
- *Fewer* lines written means *more* bad habits

Results: Lines vs. Number of Habits

- $r = -0.177$ for Q1-5 and $r = -0.462$ for Q1-10
- *Fewer* lines written means *more* bad habits

- Recall: negative correlation between confidence and bad habits
- *More* confident students means *fewer* bad habits

Results: Lines vs. Number of Habits

- Correlation between confidence and number of lines written
- *r = 0.159* for Q1-5 and *r = 0.300* for Q1-10
- *More* confident students write *more* code

Performance Analysis

- Does the previous result mean longer code is better?
- Individually marked all submissions
- Checked correlation between mark and length, mark and confidence

Performance Analysis

- Positive correlation between lines written and mark received
  - r = 0.314 for Q1-5 and r = 0.541 for Q1-10

	Q1	Q2	Q3	Q4	Q5
r	-0.051	0.107	0.168	0.343	0.472
	Q6	Q7	Q8	Q9	Q10
r	0.355	0.231	0.749	0.576	-0.250

Performance Analysis

- Positive correlation between confidence and mark received
  - r = 0.300 for Q1-5 and r = 0.332 for Q1-10

	Q1	Q2	Q3	Q4	Q5
r	0.160	0.230	0.081	0.298	0.289
	Q6	Q7	Q8	Q9	Q10
r	-0.096	0.538	0.042	0.620	-0.020

Performance Analysis

More confident students write *more* code which receives a *higher* mark

- Directly contradicts the results of an earlier study
  - “The best code is not always the longest”
  - Cindy Norris, Frank Barry, James B. Fenwick Jr., Kathryn Reid, and Josh Rountree. 2008. ClockIt: collecting quantitative data on how beginning software developers really work. SIGCSE Bull. 40, 3 (June 2008), 37-41

Future Work

- Usability study of the Code Visualizer
- Expand list of habits
- Expand to other languages
- Cluster analysis

Acknowledgements

- Angie Pinchbeck
- Abdallah Mohamed
- Norbert Eke
- Robert Hill