The Frankenstein Calculator

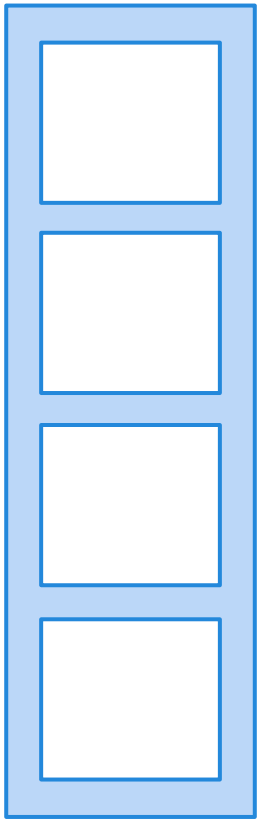
Albert Cui, Kenneth Hungria, Dan Schlosser

Introduction

- Goals
- Calculator usage
- The platform
- Software architechture
- Software details
- Lessons learned
- Criticism

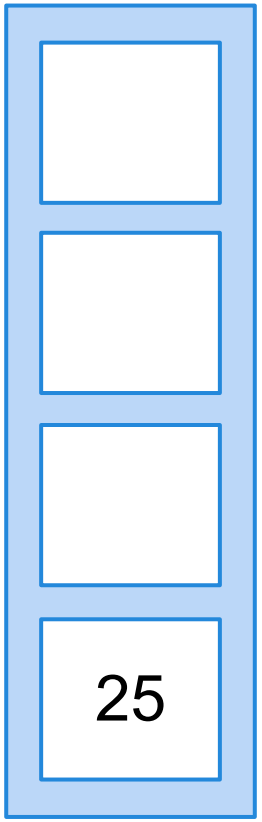
User Guide: Adding to the stack

1. [2]
2. [5]
3. [INPUT]



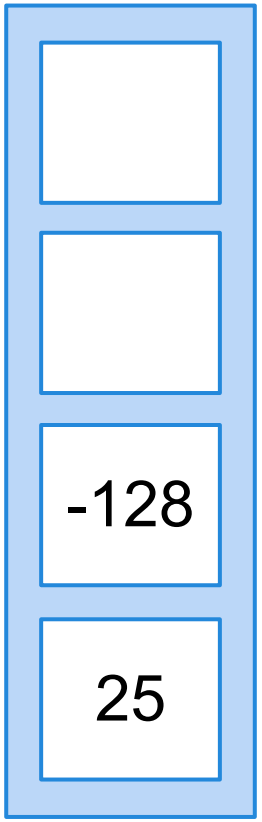
User Guide: Adding to the stack

1. [2]
2. [5]
3. [INPUT]
4. [1]
5. [+/-]
6. [2]
7. [8]
8. [INPUT]



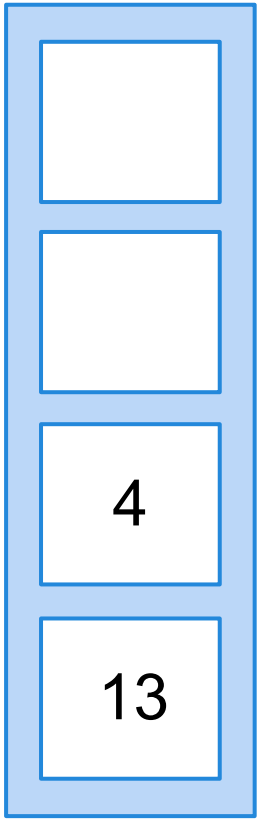
User Guide: Adding to the stack

1. [2]
2. [5]
3. [INPUT]
4. [1]
5. [+/-]
6. [2]
7. [8]
8. [INPUT]



User Guide: Operations

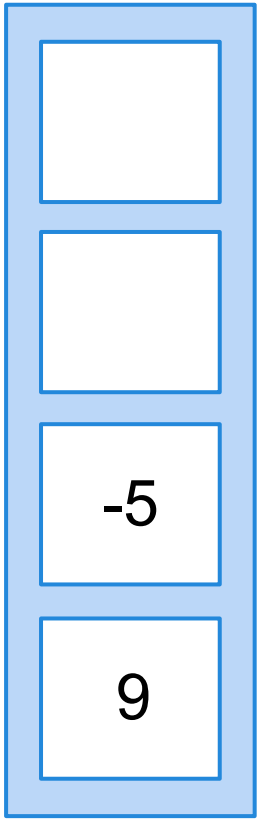
1. [1]
2. [3]
3. [INPUT]
4. [4]
5. [-]



-5(13 - 4)

User Guide: Operations

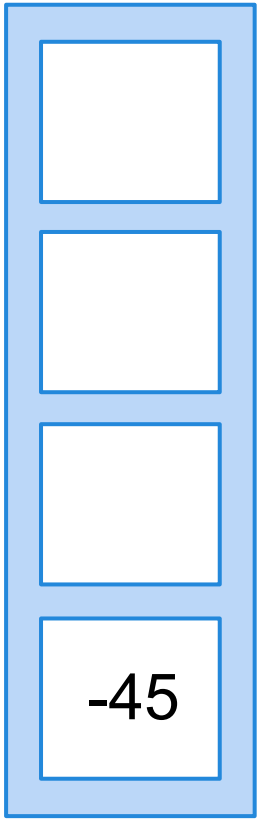
1. [1]
2. [3]
3. [INPUT]
4. [4]
5. [-]
6. [+/-]
7. [5]
8. [*]



-5(13 - 4)

User Guide: Operations

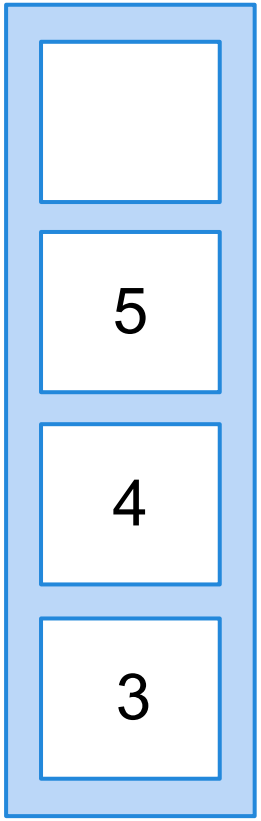
1. [1]
2. [3]
3. [INPUT]
4. [4]
5. [-]
6. [+/-]
7. [5]
8. [*]



-5(13 - 4)

User Guide: Operations

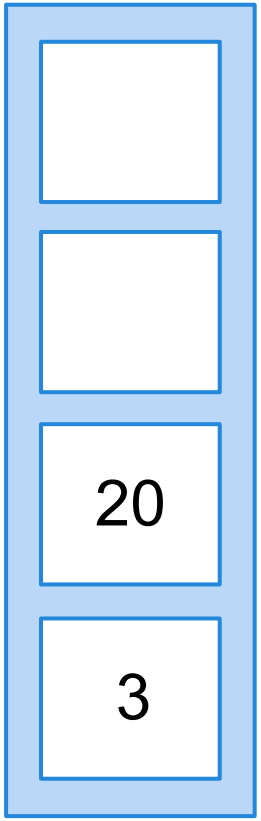
1. [3]
2. [INPUT]
3. [4]
4. [INPUT]
5. [5]
6. [*]



3 - (4*5)

User Guide: Operations

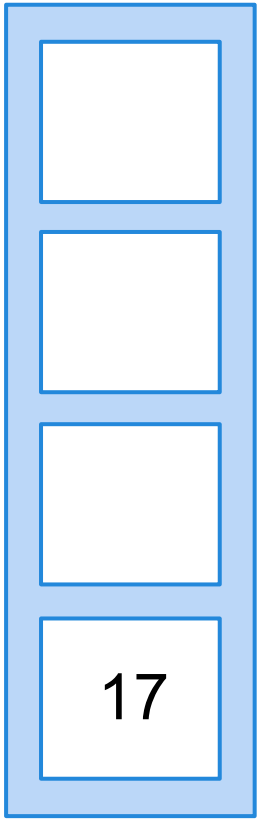
1. [3]
2. [INPUT]
3. [4]
4. [INPUT]
5. [5]
6. [*]
7. [-]



3 - (4*5)

User Guide: Operations

1. [3]
2. [INPUT]
3. [4]
4. [INPUT]
5. [5]
6. [*]
7. [-]



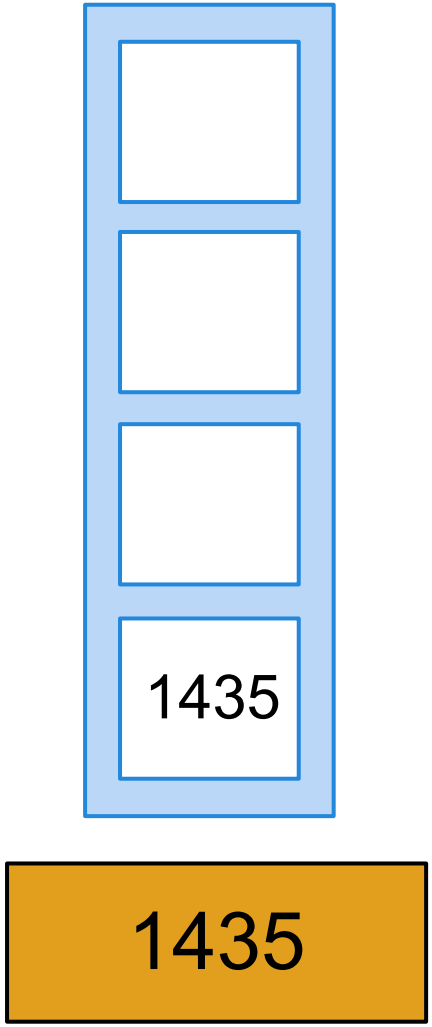
3 - (4*5)

User Guide: Backspace

1. [8]
2. [←]
3. [←]

⋮

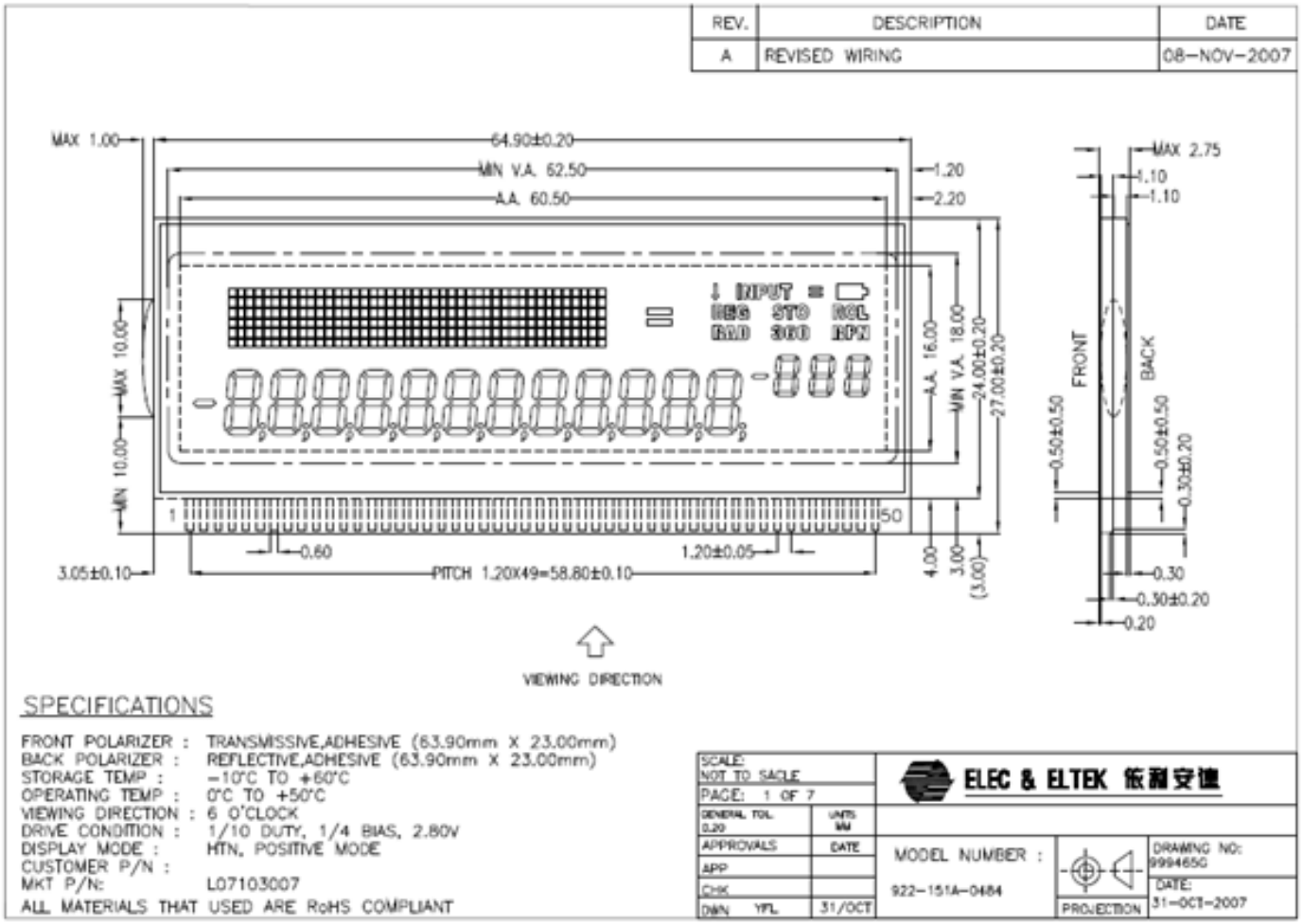
4. [←]
5. [1]
6. [7]
7. [←]
8. [4]
9. [3]
10. [5]
11. INPUT



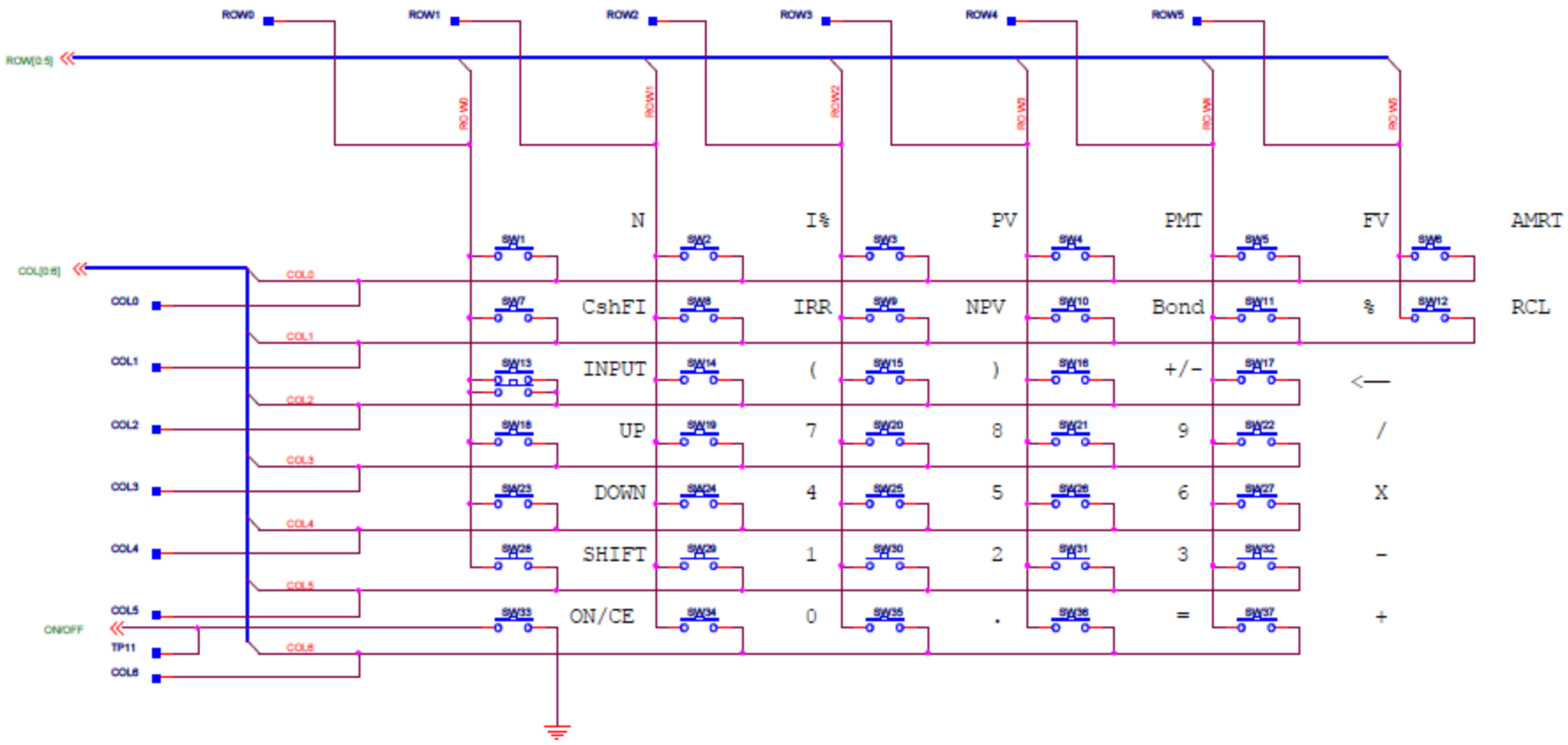
The Platform

- Amtel 30MHz, low-power AT91SAM7L128 ARM7 chip
- LCD Display
  - 2 lines
  - 8 character scrolling + indicators
  - 12 digits + 3
- Keyboard
  - 37 keys + on/off key
- JTAG connector

The LCD Display



The Keyboard



Provided Library Functions

- keyboard_init
- lcd_int
- lcd_put_char7
  - char, col
- lcd_print7
  - Prints a string of characters starting from the left
- keyboard_column_high and keyboard_column_low
- keyboard_row_read

Software Architecture

4 Parts

1. Function to print integers on the LCD
2. Reading the keyboard
3. Understanding and displaying the inputs
4. Calculations

Software Details: Lab 1

```
void displayInt(int num) {
        clearScreen();
        int index = 11
        isNeg = 0;
        if (num < 0)
                isNeg = 1;
                num *= -1
        if(num == 0) //special case, avoid infinite loop
                lcd_put_char7('0', index);
        while (num > 0) {  //Displays characters from right to lef
                char c = (char)(num \% 10 + '0'); // integer -> char
                lcd_put_char7(c, index); //displays the char
                index--;
                num /= 10;
        }
        if(isNeg == 1)
                lcd_put_char7('-', index); //show the '-'
}

void clearScreen() {  //clears the screen
        int x;
        for(x = 0;x<12;x++){
                lcd_put_char7(' ',x);
}
```

Software Details: Lab 2

```
for (;;) {
        char keyVal = keyboard_key();
        if(keyVal != 0) {
        lcd_clear();
        lcd_put_char7(keyVal,0);
}

char keyboard_key() {
        int col;
        int row;
        int notCol = 0;
        for(col = 0; col <= 6; col++) { // For each column
                keyboard_column_low(col); //set column low
                for(notCol = 0; notCol <=6; notCol++) {
                        if(notCol != col)
                                keyboard_column_high(notCol); //set all other columns high
                }
                for(row=0;row<=5;row++) { // A row high => button pressed
                        if(!keyboard_row_read(row))
                                return arr[col][row]; // Returns the assigned char
                }
        }
}
```

Software Details: Lab 2 cont.

```
const char arr[NUM_COLUMNS][NUM_ROWS] = { // Char for each button
        {'N', 'I', 'P', 'M', 'F', 'A'},
        {'C', 'R', 'V', 'B', '\%', 'L'},
        {'U', '(', ')', '_', '<', 0},
        {'^', '7', '8', '9', '/', 0},
        {'~', '4', '5', '6', '*', 0},
        {'S', '1', '2', '3', '-', 0},
        {0, '0', '.', '=', '+', 0}};
```

Software Details: Lab 3

```
void keyboard_get_entry(struct entry *result) {
	int isNegative = 0; // boolean negative indicator
	int numPressed = INT_MAX; // |number pressed|, INT_MAX~null
	int released = 1; // 1 if no button is being pressed
	int keyVal=-1;

	while (keyVal!='+' && keyVal!='-' &&keyVal!='/' && keyVal!='*' && keyVal!='\r') { // Not an operation
		keyVal = keyboard_key(); // Read from keyboard
		if (keyVal != -1 && released == 1) { // count each press once
			if (keyVal >= '0' && keyVal<='9' && (numPressed==INT_MAX || numPressed+1<INT_MAX/10)){ // number
				numPressed = (numPressed==INT_MAX)? keyVal-'0' :numPressed*10+keyVal-'0'; // signed
				lcd_print_int(numPressed*((isNegative)?-1:1)); // signed
			} else if(keyVal=='~') { // The user pressed the +/- key.
				isNegative = (isNegative)? 0:1; // flip negative
				lcd_clear();
				if(numPressed==INT_MAX)
					lcd_put_char7('-',LCD_NUM_REAL_COLUMNS-1);
				else {
					lcd_print_int(numPressed*((isNegative)?-1:1));
				}
```

Software Details: Lab 3 cont.

```
                        } else if(keyVal=='\b') { // The user pressed the backspace key.
                                if (numPressed == INT_MAX && isNegative) {
                                        isNegative = (isNegative)? 0:1; // flip negative
                                        lcd_clear();
                                } else if (numPressed < 10) {
                                        lcd_clear();
                                        if(isNegative){lcd_put_char7('-',LCD_NUM_REAL_COLUMNS-1);}
                                        numPressed = INT_MAX; // Reset numPressed
                                } else if (numPressed != INT_MAX) {
                                        numPressed = numPressed / 10; // remove ones digit
                                        lcd_print_int(numPressed*((isNegative)?-1:1)); // signed
                                }
                        }
                        released = 0; //The user has begun holding down a key.
                } else if (keyVal == -1) { released = 1;}
        }
        result->operation=keyVal;
        result->number=numPressed*((isNegative)?-1:1);
}
```

Software Details: Lab 3 cont.

```
const char keyboard_keys[NUM_COLUMNS][NUM_ROWS] = {
 {'N',  'I', 'P', 'M', 'F', 'A'},
 {'C',  'R', 'V', 'B', '%', 'L'},
 {'\r', '(', ')', '~', '\b', 0},
 {'\v', '7', '8', '9', '/',  0},
 {'\n', '4', '5', '6', '*',  0},
 {'S',  '1', '2', '3', '-',  0},
 { 0,   '0', '.', '=', '+',  0}};
```

Software Details: Lab 4

```
int push(int element){
   if (top == mysize)
      return INT_MAX;
   top++;
   stack[top] = element;
   return element;
}

int pop() {
   if (top == -1)
      return INT_MAX;
   int element = stack[top];
   top --;
   return element;
}

int peek() {
   return push(pop());
}
```

Software Details: Lab 4 cont.

```
for(;;){
      keyboard_get_entry(&entry);
      if(entry.number != INT_MAX)
         push(entry.number);
      if(top >= 1){
         if(entry.operation == '+')
            push(pop()+pop());
         else if(entry.operation == '-'){
            int first = pop();
            int second = pop();
            push(second - first);
         }else if(entry.operation == '*')
            push(pop()*pop());
      }
      lcd_print_int(peek());
   }
```

Lessons Learned

- Working in a team
- Expansion of coding techniques
- Balancing workload
- Think simple
- Mix of skills work well

Criticism of the Course

- Lab was a good simulation of a group project
- Not much coding necessary, mostly implementations
- Code reviews helped with implementation
- "Malicious" coding?