

EECS 10: COMP METHODS IN ECE

Discussion 7

Guantao Liu
guantaol@uci.edu

07/14/2015

Assignment Discussion

- Assignment 4
 - A menu-driven calculator for floating point numbers
 - Addition, Subtraction, Square root, ...

Tuesday

1. Add a floating point number to the current result;
2. Subtract a floating point number from the current result;
3. Multiply the current result by a floating point number;
4. Divide the current result by a floating point number;
5. Take the absolute value of the current result;

6. Get the approximate square root of the current result;
7. Get the sine of the current result;
8. Get the cosine of the current result;
9. Get the tangent of the current result;
10. Get the approximate Nth root of the current result;

11. Quit.

Thursday

Program Interface

```
Welcome to my floating point number calculator!  
Please input a floating point number: 4.2
```

Prompt

```
-----  
The current result is: 4.200000  
1. Add a floating point number to the current result;  
2. Subtract a floating point number from the current result;  
3. Multiply the current result by a floating point number;  
4. Divide the current result by a floating point number;  
5. Take the absolute value of the current result;  
6. Get the approximate square root of the current result;  
7. Get the sine of the current result;  
8. Get the cosine of the current result;  
9. Get the tangent of the current result;  
10. Get the approximate Nth root of the current result;  
11. Quit.
```

```
Please enter a selection:
```

Repetition

EECS 10 Discussion 7, July, 2015

(c) 2015 Guantao Liu

3

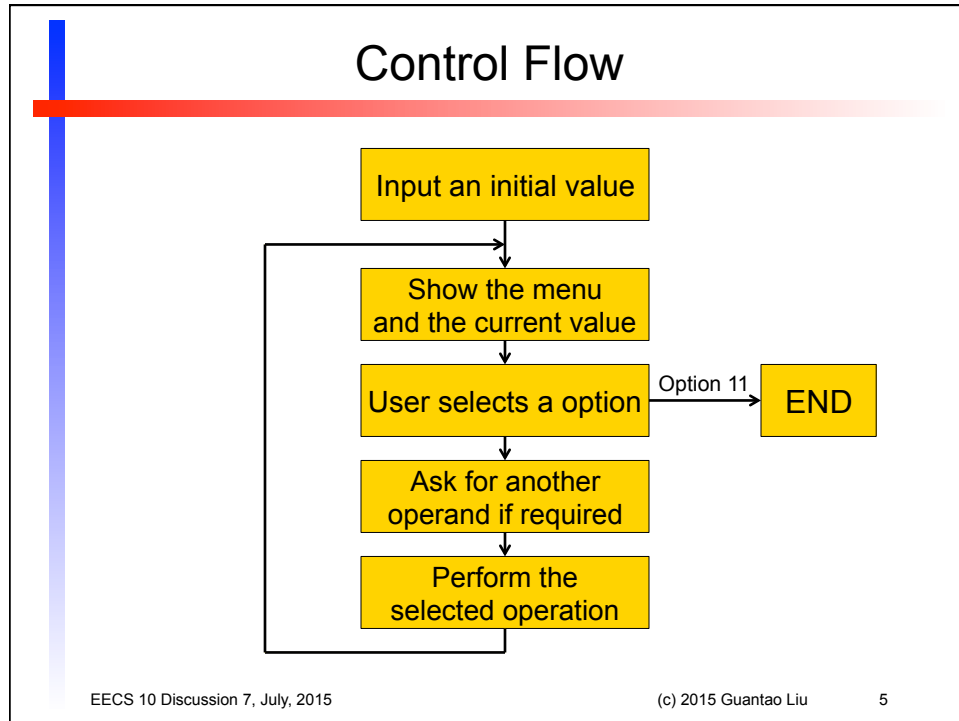
Program Implementation

- The program exits only when the user chooses to quit (Option 11).
 - Use a loop structure to repeat menu and operations
- Ask for additional operand if needed:
 - Addition, subtraction and etc. : require a second operand
 - Absolute, sine, cosine and etc. : only need the current result value
- Use `double` to hold double precision floating point numbers

EECS 10 Discussion 7, July, 2015

(c) 2015 Guantao Liu

4



Functions

- Define every operation as an individual function

```

/* Add two floating point numbers and return the value */
double Add(double op1, double op2) {
    ...;
}
...;
/* Calculate the absolute of the input value */
double Abs(double op1) {
    ...;
}
...;
/* Calculate the approximate square root */
double ApproximateRoot2(double op1) {
    ...;
}
  
```

EECS 10 Discussion 7, July, 2015 (c) 2015 Guantao Liu 6

Function Declarations

- Function Declaration
 - Function name, parameters and return type
 - Declarations must be before function calls.
- Use following function declarations for operations in the menu-driven calculator


```
double Add(double op1, double op2);
double Subtract(double op1, double op2);
double Multiply(double op1, double op2);
double Divide(double op1, double op2);
double Abs(double op1);
```
- Invoke a function when the user selects a corresponding option.

EECS 10 Discussion 7, July, 2015

(c) 2015 Guantao Liu

7

Function Definitions

- Function definition
 - An implicit function declaration
 - A function can be defined only once in a program.
- Program example: **Cylinder.c** (pp. 5 of Lecture 6)

```
double Surface(double r, double h)
{
    double side, lid;

    side = CirclePerimeter(r) * h;
    lid = CirclePerimeter(r);

    return (side + 2*lid);
}
```

```
double Add(double op1, double op2) {
    ...;
}
```

EECS 10 Discussion 7, July, 2015

(c) 2015 Guantao Liu

8

Function Calls

- Function call
 - Invoke a function
 - Supply arguments for formal parameters
- Call a function with arguments

```
double Add(double op1, double op2)
{
    ...;
}

int main()
{
    double currValue; // current result
    double newValue; // a second operand

    ...;
    currValue = Add(currValue, newValue);
    ...;
}
```

Actual Arguments

EECS 10 Discussion 7, July, 2015

(c) 2015 Guantao Liu

9

Assignment Discussion

- Error handling
 - Print out an error message when the operand is invalid.
 - e.g. "ERROR: Division by zero!" when the divisor is 0.
 - Prompt the user until a proper divisor is entered.

```
...
Please enter a selection: 4
-----
Please input a floating point number operand: 0
ERROR: Division by zero!
Please input a floating point number operand:
```

- Verify your program with Step 1-9 on Page 6
- Name your files as **calculator.c**, **calculator.txt** and **calculator.script**

EECS 10 Discussion 7, July, 2015

(c) 2015 Guantao Liu

10