C++ Programming
Selection Practice

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Simple Calculator

- Given two **numbers** and a sign between them which will indicate if the user want the **addition, subtraction, division or multiplication** of these two numbers, find the value of the answer.
- Inputs ⇒ outputs
  - 7 + 55 ⇒ 62
  - 7 * 10 ⇒ 70
- **Stop** the video for a few minutes, and sketch some code

Simple Calculator

```cpp
int main() {
    // Good choice for a number here is double
    double num1, num2;
    char operation;

    cin >> num1 >> operation >> num2;

    if (operation == '+')
        cout << num1 + num2 << "\n";

    else if (operation == '-')
        cout << num1 - num2 << "\n";

    else if (operation == '*')
        cout << num1 * num2 << "\n";

    else
        cout << num1 / num2 << "\n";

    return 0;
}
```

Console | Problems | Tasks | Properties | Call Graph

<terminated> ztemp [C/C++ Application] /home/moustafa/workspa

```
3 * 6
18
```

Minimum of 2 numbers

- Read 2 **integers** and print the minimum one of them
- Inputs ⇒ outputs
  - 10 20 ⇒ 10
  - 70 5 ⇒ 5
- Stop the video for a few minutes, and sketch some code

Minimum of 2 numbers

```
4  int main() {
5      int num1, num2;
6
7      cin >> num1 >> num2;
8
9      if (num1 < num2)
10         cout << num1 << "\n";
11     else
12         cout << num2 << "\n";
13
14     return 0;
15 }
```

Console | Problems | Tasks | Propert

<terminated> ztemp [C/C++ Application] /home/m

```
20 10
10
```

Minimum of 3 numbers

- Read 3 **integers** and print the minimum one of them
- Inputs
  - 10 20 30 ⇒ 10
  - 70 5 15 ⇒ 5
- Stop the video for a few minutes:
  - Think what all cases setups we need to make sure code is correct?
  - Sketch the code. There are many ways to code it!

Min of 3 numbers: Way #1

```
int main() {
    int num1, num2, num3;

    cin >> num1 >> num2 >> num3;

    if (num1 < num2) {
        // Then either num1 or num3 is the answer
        if (num1 < num3)
            cout << num1 << "\n";
        else
            cout << num3 << "\n";
    } else  // Then either num2 or num3 is the answer
    {
        if (num2 < num3)
            cout << num2 << "\n";
        else
            cout << num3 << "\n";
    }

    return 0;
}
```

Console | Problems | Tasks | Properties | Call Graph

<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces

```
20 10 30
10
```

Min of 3 numbers: Way #2

```
16 int main() {
17     int num1, num2, num3;
18
19     cin >> num1 >> num2 >> num3;
20     // Code FIX: use <= NOT <
21     if (num1 <= num2 && num1 <= num3)
22         cout << num1 << "\n";
23     else if (num2 <= num1 && num2 <= num3)
24         cout << num2 << "\n";
25     else
26         cout << num3 << "\n";
27
28     return 0;
29 }
```

Min of 3 numbers: Way #3

```
4  int main() {
5      int num1, num2, num3;
6
7      cin >> num1 >> num2 >> num3;
8
9      int answer = num1;
10
11     if (answer > num2)
12         answer = num2;
13
14     if (answer > num3)
15         answer = num3;
16
17     cout << answer << "\n";
18
19     return 0;
20 }
21
```

Console ⊠ Problems Tasks Prop

<terminated> ztemp [C/C++ Application] /home

```
2 3 1
1
```

Is even? Print digits

- Read an integer N, then do the following
  - If the number is even: **print** last digit of it
  - If the number is odd: do following:
    - If number < 1000, **print** last 2 digits
    - If number > 1000 and number < 1000000, **print** last 4 digits
    - Otherwise, **print** its negative value
- Stop the video and think: 1) Code 2) Good tests
- Testing examples of good coverage:
  - 234 ⇒ even ⇒ 4
  - 157 ⇒ 57
  - 567169 ⇒ 7169
  - 1000001 ⇒ -1000001

Is even? Print digits

```
4  int main() {
5      int num;
6      cin >> num;
7
8      bool is_even = (num % 2 == 0);
9
10     if (is_even)
11         cout << num % 10 << "\n";
12     else {
13         if (num < 1000)
14             cout << num % 100 << "\n";
15         else if (num < 1000000)
16             cout << num % 10000 << "\n";
17         else
18             cout << -num << "\n";
19     }
20     return 0;
21 }
22
23
```

Console | Problems | Tasks | Properties | Cal

<terminated> ztemp [C/C++ Application] /home/moustafa/w

```
567169
7169
```

- Recall %2 can be used to know if number is even
  - 0 ⇒ even
  - 1 ⇒ odd
- Notice we have if for even, then else for odd
- This else has big body for handling the 3 odd cases

Last 3 digits!

- Read an integer and do the following:
  - If number < 10000, **say** this is a small number
  - Otherwise Sum the last 3 digits of the number
    - If the sum is odd, **say** this is a great number
    - Otherwise, If sum is even:
      - If any digit of the last 3 is odd, **say** this is a good number
      - Otherwise, **say** this is a bad number
- Stop the video and think: 1) Code 2) Good tests
  - Be a good tester:
  - Find all needed test cases that covers all possible scenarios

Last 3 digits!

```
int main() {
    int n;
    cin>>n;

    if (n < 10000)
        cout<<"this is a small number\n";
    else
    {
        int digit1 = n%10;
        n = n/10;
        int digit2 = n%10;
        n = n/10;
        int digit3 = n%10;  // old value of n gone

        int sum = digit1+digit2+digit3;

        if ((sum%2) != 0)   // odd
            cout<<"this is a great number\n";
        else
        {
            bool is_digit1_odd = (digit1 % 2 == 1);
            bool is_digit2_odd = (digit2 % 2 == 1);
            bool is_digit3_odd = (digit3 % 2 == 1);

            if (is_digit1_odd || is_digit2_odd || is_digit3_odd)
                cout<<"this is a good number\n";
            else
                cout<<"this is a bad number\n";
        }
    }
```

- Test cases:
  - 100
  - 10111
  - 10330
  - 10303
  - 10033
  - 10000

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."