COMPUTER CONTEST LEVEL 1

David Lu

About Me

- David Lu
- Software Engineering @ University of Waterloo, class of 2020
- Full Stack Developer @ Mattermost Inc.
- Fun facts:
  - *My dream car is a F-150 pickup truck*
  - *Dogs are adorable*
  - *I drink way too much coffee*

Course Outline

- Learning basic algorithms, and techniques to succeed in programming contests
- Topics covered:
  - *Conditionals && Loops*
  - *Arrays && Sorting*
  - *Recursion && Strings*
  - *Structures*
  - *Graph Theory*
- Requirements:
  - *Basic programming knowledge (Java/C++)*
  - *Working laptop*

Evaluations

- Weekly Homework Assignments
  - *5% each, 40% total*
  - *Submitted online*
- Biweekly Quizzes
  - *5% each, 20% total*
  - *Submitted online*
- Monthly Tests
  - *20% each, 40% total*
  - *Submitted online*
- All feedback will be provided online

Notes/Homework/Discussion

- olympiads-david.slack.com
- Slack is a team-based messaging service
- Notes will be posted in **#computer-contest** prior to each class
- Homework must be submitted to **@homeworkbot** as a Direct Message
- Discussion can occur in **#general-discussion** or **#computer-contest**

Canadian Computing Competition

- Programming competition held annually
  - *Late February to early March*
- February 22, 2017
- https://cemc.math.uwaterloo.ca/contests/contests.html

Stages

- Stage 1 (CCC)
  - *Junior and Senior*
  - *5 questions worth 15 marks each*
  - *3 hours*
- Stage 2 (CCO)
  - *Top 20 of Senior*
  - *Week-long stay at the University of Waterloo*
  - *Top 8 get to go to IOI (International Olympiad in Informatics)*

Junior

- Reasonably easy
  - *Grade 11/12 computer science material*
- Loops and conditions
- Basic algorithms
  - *Recursion*
  - *Sorting*
  - *Searching*

Questions 1 and 2	Straightforward (e.g., basic loops and conditions)
Questions 3 and 4	More challenging (e.g. some combination of loops, conditions and counting)
Question 5	Some advanced material (e.g., recursion, efficient sorting, clever algorithms)

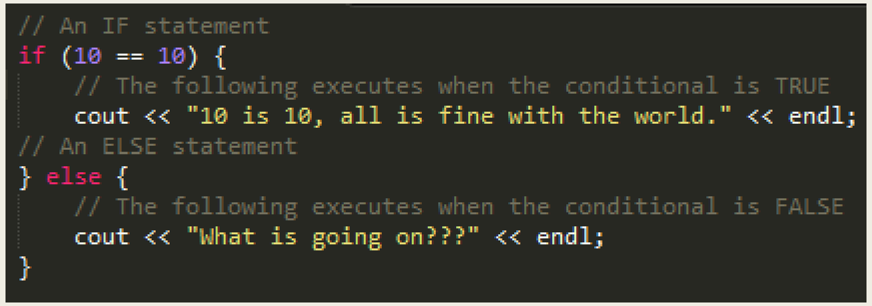
Senior

- Relatively hard
  - *Knowledge beyond grade 12 computer science*
- Basic and advanced algorithms
  - *Sorting*
  - *Searching*
  - *Recursion*
  - *Dynamic Programming*
  - *Math*
- To reach stage 2
  - *55-65 out of 75 required*

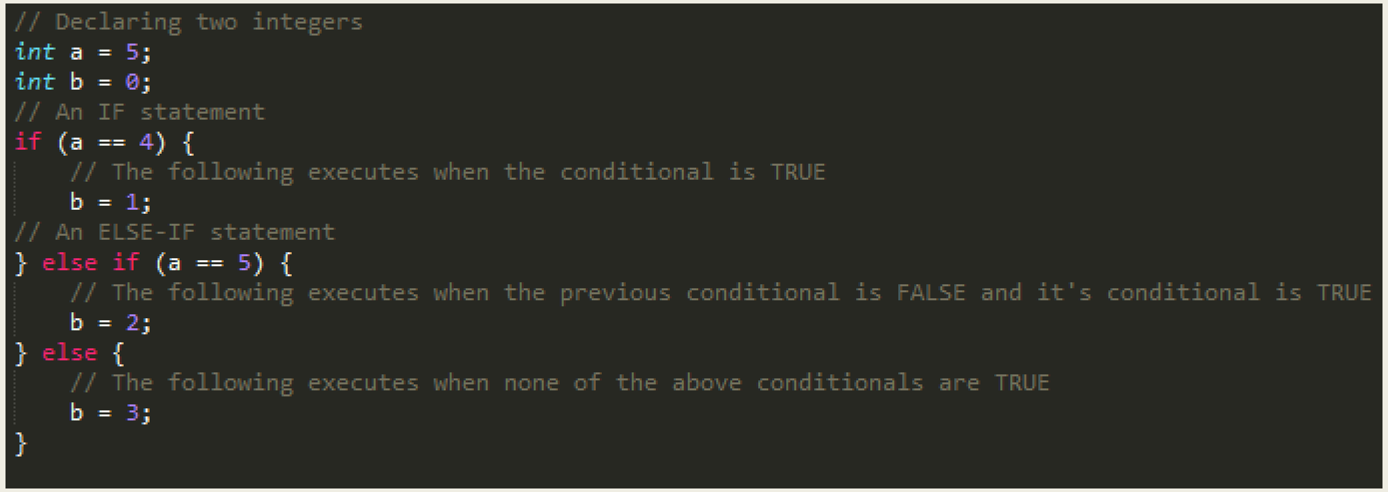
Questions 1 and 2	Basic algorithms (e.g., sorting, searching)
Questions 3 and 4	More advanced algorithms (e.g., careful counting, some mathematical reasoning)
Question 5	e.g., IOI level question

Conditionals

- Statements returning either TRUE or FALSE (Boolean)
- Can be used in if-else blocks to check for a specific condition
- The == operator is commonly used, it tests for equality



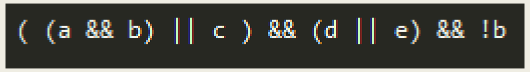
```
// An IF statement
if (10 == 10) {
    // The following executes when the conditional is TRUE
    cout << "10 is 10, all is fine with the world." << endl;
// An ELSE statement
} else {
    // The following executes when the conditional is FALSE
    cout << "What is going on???" << endl;
}
```



```
// Declaring two integers
int a = 5;
int b = 0;
// An IF statement
if (a == 4) {
    // The following executes when the conditional is TRUE
    b = 1;
// An ELSE-IF statement
} else if (a == 5) {
    // The following executes when the previous conditional is FALSE and it's conditional is TRUE
    b = 2;
} else {
    // The following executes when none of the above conditionals are TRUE
    b = 3;
}
```

Boolean Operators

- Relational: == (equals to), >, <, >=, <=
- Logical: && (AND), || (OR)
- Negation: ! (NOT), != (not equals to)
- Operators may be combined using braces/other operators
- All operators require 2 inputs except for NOT
- Evaluation
  - *AND is TRUE when both inputs are TRUE*
  - *OR is only FALSE when both inputs are FALSE*
  - *NOT is FALSE when it’s input is FALSE*



```
( (a && b) || c ) && (d || e) && !b
```

If-Else Blocks

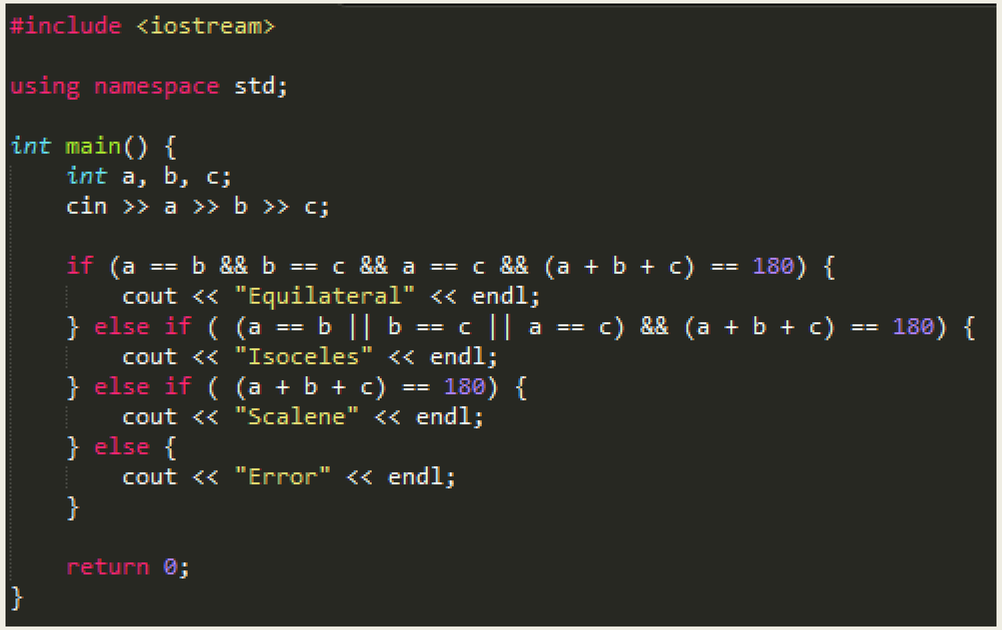
- An if-else block consists of
  - *One IF statement*
  - *Multiple ELSE-IF statements (optional)*
  - *One ELSE statement (optional)*
- The conditionals are evaluated sequentially
- Once a conditional is evaluated to be true, the rest of the block stops execution
  - *If an ELSE-IF statement is executed, the previous conditionals must be FALSE*
- If no conditionals are true, the ELSE statement is executed

Practice – Triangle Times

- Given three angles of a triangle from user input, output:
  - *“Equilateral” if all angles are 60 degrees*
  - *“Isosceles” if the angles add up to 180 degrees and exactly 2 are the same*
  - *“Scalene” if the angles add up to 180 degrees and no angles are the same*
  - *“Error” if the angles do not add up to 180 degrees*
- Solve this 2 ways
  - *Only using IF statements*
  - *Using IF-ELSE blocks*

Solution

- See `class1_ex1.cpp`



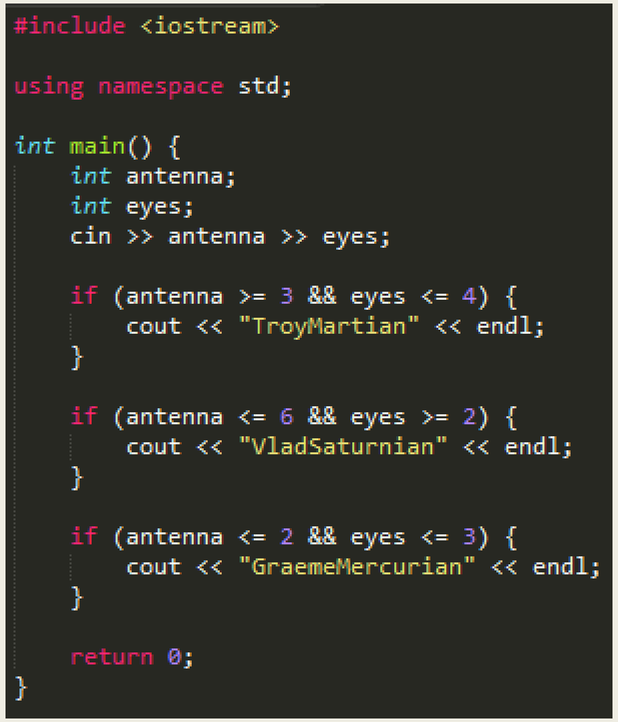
```cpp
#include <iostream>

using namespace std;

int main() {
    int a, b, c;
    cin >> a >> b >> c;

    if (a == b && b == c && a == c && (a + b + c) == 180) {
        cout << "Equilateral" << endl;
    } else if ( (a == b || b == c || a == c) && (a + b + c) == 180) {
        cout << "Isoceles" << endl;
    } else if ( (a + b + c) == 180) {
        cout << "Scalene" << endl;
    } else {
        cout << "Error" << endl;
    }

    return 0;
}
```

Practice - Aliens

- NASA has classified aliens based on their antenna and eye counts
  - *TroyMartian: at least 3 antenna, at most 4 eyes*
  - *VladSaturnian: at most 6 antenna, at least 2 eyes*
  - *GraemeMercurian, at most 2 antenna, at most 3 eyes*
- Given the number of antenna and eyes of an aliens, output all possible types

Solution

- See `class1_ex2.cpp`

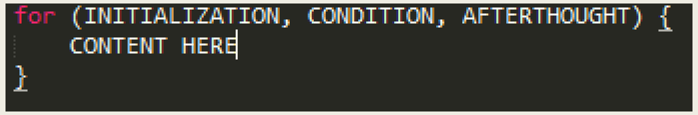


```
#include <iostream>

using namespace std;

int main() {
    int antenna;
    int eyes;
    cin >> antenna >> eyes;

    if (antenna >= 3 && eyes <= 4) {
        cout << "TroyMartian" << endl;
    }

    if (antenna <= 6 && eyes >= 2) {
        cout << "VladSaturnian" << endl;
    }

    if (antenna <= 2 && eyes <= 3) {
        cout << "GraemeMercurian" << endl;
    }

    return 0;
}
```

Loops

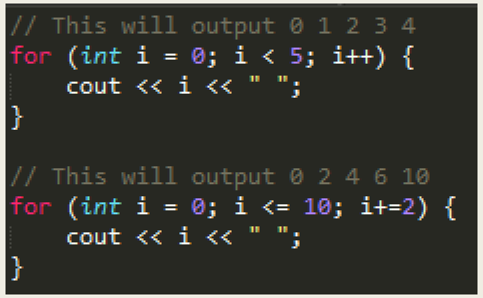
- Loops are used to repeat blocks of code in **iterations**
- There are three types of loops
  - *For*
  - *While*
  - *Do-While*
- In general, all types of loops are interchangeable

For Loop



```
for (INITIALIZATION, CONDITION, AFTERTHOUGHT) {
    CONTENT HERE
}
```

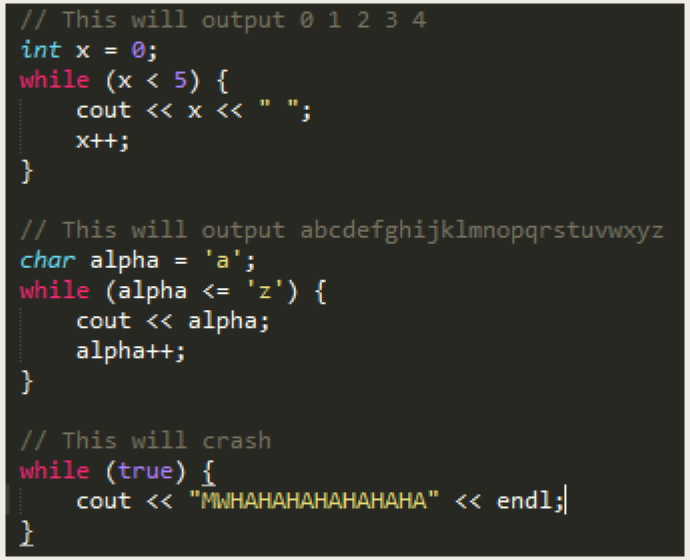
- Three parts to a for loop
  - *Initialization, any statement needed to set up the loop (e.g.* `int i = 0;`*)*
  - *Condition, the loop runs while this condition is TRUE (e.g.* `i < 10;`*)*
  - *Afterthought, what runs after every iteration of the loop (e.g.* `i++`*)*
- A loop iteration looks like: BODY => AFTERTHOUGHT => CONDITION
- Examples



```
// This will output 0 1 2 3 4
for (int i = 0; i < 5; i++) {
    cout << i << " ";
}

// This will output 0 2 4 6 10
for (int i = 0; i <= 10; i+=2) {
    cout << i << " ";
}
```

While Loop

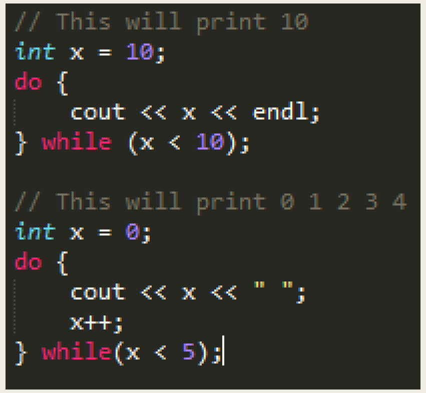
- Runs **while** a condition is TRUE
- A loop iteration looks like: CONDITION => BODY
- Examples:



```
// This will output 0 1 2 3 4
int x = 0;
while (x < 5) {
    cout << x << " ";
    x++;
}

// This will output abcdefghijklmnopqrstuvwxyz
char alpha = 'a';
while (alpha <= 'z') {
    cout << alpha;
    alpha++;
}

// This will crash
while (true) {
    cout << "MWHAHAHAHAHAHAHA" << endl;
}
```

Do-While Loops

- Exactly the same as a while loop
- The condition is checked last, instead of first
  - *Will always run once*
- A loop iteration looks like BODY => CONDITION



```
// This will print 10
int x = 10;
do {
    cout << x << endl;
} while (x < 10);

// This will print 0 1 2 3 4
int x = 0;
do {
    cout << x << " ";
    x++;
} while(x < 5);
```

Which Loop Do I Use?

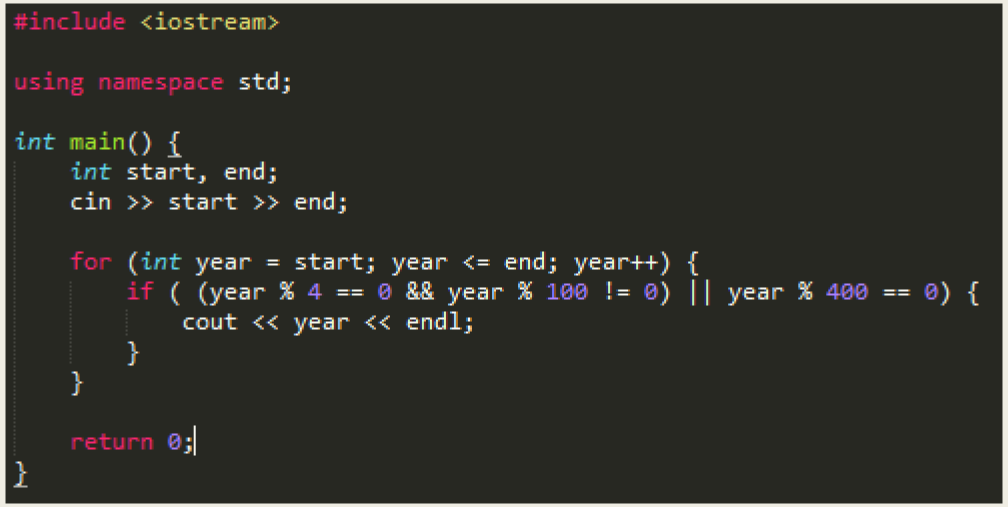
- For: When the number of iterations is known
- While: When the conditions are known, but not the number of iterations
- Do-While: When something must be done prior to checking the conditions
  - *Often used for one-off things such as input initialization*
- All loop examples are found in `class1_ex3.cpp`

Practice – Leap Years

- Given a start year and an end year, output all leap years between the two years (inclusive)
- A leap year is defined as a year that is divisible by 4 and not by 100, or by 400

Solution

- See `class1_ex4.cpp`



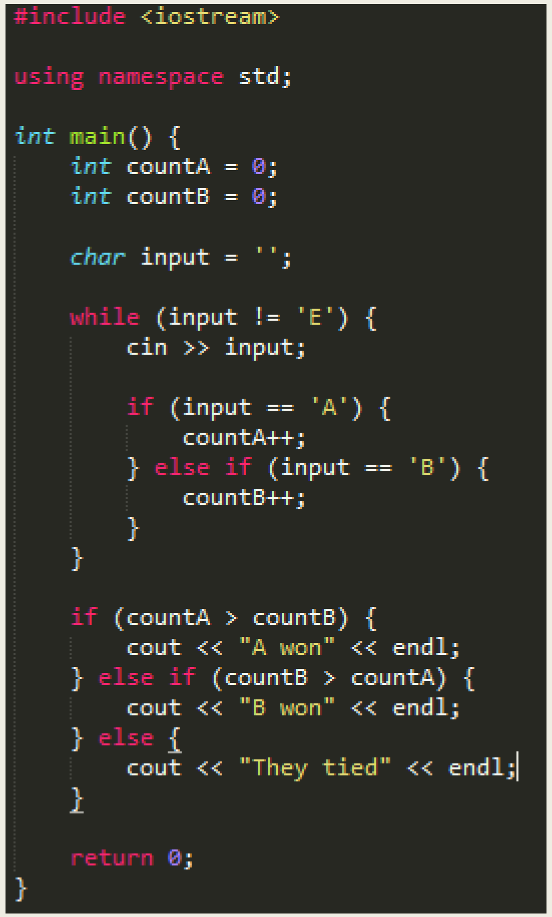
```cpp
#include <iostream>

using namespace std;

int main() {
    int start, end;
    cin >> start >> end;

    for (int year = start; year <= end; year++) {
        if ( (year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
            cout << year << endl;
        }
    }

    return 0;
}
```

Practice – Vote Counting

- A vote was held to determine the winner of a contest
- The votes will be denoted as either A or B
- Your program should take a list of votes, and determine the outcome
- The end of input will be denoted by the character E
- Output whether A won, B won, or if it was a tie

Solution

- See `class1_ex5.cpp`



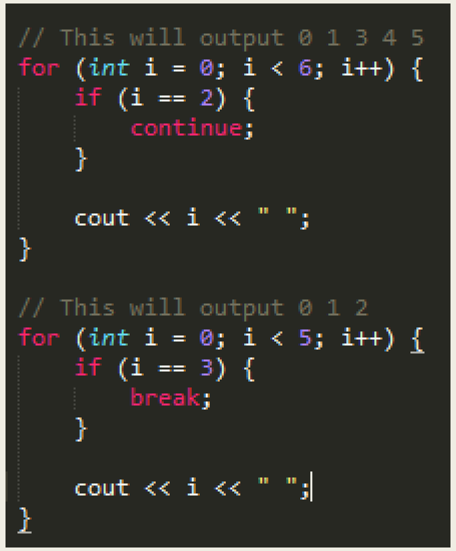
```
#include <iostream>

using namespace std;

int main() {
    int countA = 0;
    int countB = 0;

    char input = '';

    while (input != 'E') {
        cin >> input;

        if (input == 'A') {
            countA++;
        } else if (input == 'B') {
            countB++;
        }
    }

    if (countA > countB) {
        cout << "A won" << endl;
    } else if (countB > countA) {
        cout << "B won" << endl;
    } else {
        cout << "They tied" << endl;
    }

    return 0;
}
```

Infinite Loops

- Some loops cannot exit due to poorly-written conditions or intentional bad behaviour
  - *Those loops crash*
- When writing an infinite loop on purpose, ensure you have a way to escape…

Breaks, Continues, and Returns

- Two keywords are built in for loops: **break** and **continue**
- Break will exit a loop
- Continue will skip to the next iteration of the loop
- Example: (see `class1_ex3.cpp`)



```
// This will output 0 1 3 4 5
for (int i = 0; i < 6; i++) {
    if (i == 2) {
        continue;
    }

    cout << i << " ";
}

// This will output 0 1 2
for (int i = 0; i < 5; i++) {
    if (i == 3) {
        break;
    }

    cout << i << " ";
}
```

Homework

- Solve these questions
  - *https://dmoj.ca/problem/ccc10j1*
  - *https://dmoj.ca/problem/ccc15j1*
  - *https://dmoj.ca/problem/ccc12j2*
  - *https://dmoj.ca/problem/ccc13j3*
- Submit solutions to @homeworkbot on Slack
  - *A single zipped file please*