


Competitive Programming

Lesson 1 – Introduction



Written by Jason Sun

Reviewed by Puneet Bhat

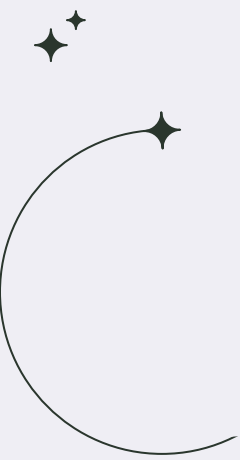
Presentation Format

This slide will be **lengthier** than the usual academic requirement, as you can always go back to these slides to **catch up** or deepen your **understanding** of any topics that you didn't quite understand.

We will be giving an **introduction** to competitive programming, as well as giving you practice on solving/submitting solutions to **coding problems** (similar to how you would do so in a contest).

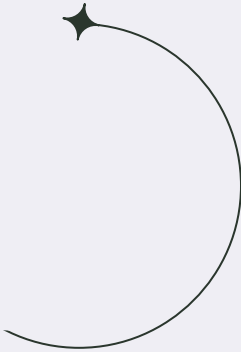


Prior Experience Inquiry



Raise your hand if you
already do competitive
programming!

- Graph Theory
- Prefix Sum Arrays
- Segment Trees
- etc



Raise your hand if you
have done any **coding**
contests before!

- Waterloo CCC
- Leetcode Contests
- etc

Why This?

Coding is easy. But **programming** is hard.

Learning competitive programming, or contest coding, can not only aid you directly in contests (like Waterloo CCC) and some coding interviews, but also in any other aspect of computer science, where you need to find the best way to solve a problem, with code!

Common Questions



Languages?

Any is fine! However, CCC allows C, C++, Python 2, Python 3, and Java (*Note: CCO only allows C++*).



Commitment

The more you practice and learn, the better the results!



About

Solving coding problems with logic, data structures, and algorithms.



Tools Needed

All you need is a browser (installed IDE is optional).

Canadian Computing Contest



About

Every year, on the third Wednesday of February.

- February 19, 2025 this year

For more details:

<https://cemc.uwaterloo.ca/contests/ccc>



Problems

Two different contests to choose from

- Junior: Easier (grade 9-10)
- Senior: Harder (grade 11-12)

Consists of 5 problems, each worth 15 points, for a total of 75 points.



Results

You can get distinction (top 25%) or honour roll

- Top ~500 for junior
- Top ~250 for senior

Waterloo invites the top 20 senior takers to CCO.

- Canadian IOI team
- You can get medals, offers, and scholarships

Question Example

Your friend Bob needs help counting.
He gives you an integer n .

Output the integers from 1 to n , all on
different lines, in ascending order.

Output needs to be exact. Example on
the right.

INPUT

4

OUTPUT

1

2

3

4



Solution Example

```
1  n = int(input())  
2  
3  for i in range(1, n+1):  
4      |   print(i)  
5
```

As you can see, you take in input, perform the required logic, and output what they want.



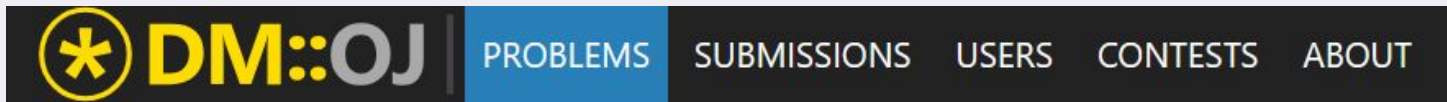
Where do we Practice?

There are many websites to read problems and submit solutions, where the website runs the code for you, and checks if your output is correct, depending on the various inputs they give to your code.

We will be using **DMOJ**, basically what all Canadian students use.

Open the website: <https://dmoj.ca/>
Sign up in the top right corner.

Now you can click the problems section to see a list of problems you can do!
There are also contests hosted on the website too.





Where do we Write Code?


You can use any IDE that you want.

After you are done writing and are ready to submit, just go to the DMOJ problem, press the **submit button**, **copy and paste** the code, **select the language** you used, and **submit** it.

My recommendation is:

<https://ide.usaco.guide/> for Python 3, Java, C++
or <https://www.onlinegdb.com/> for more languages.

I recommend your IDE having a text file for input and a text file for output, instead of a console format, since it usually makes your life easier for most problems.



Lacking Motivation?

Competition will motivate you a lot! Kind of like trying to win in any PvP game (like CoD, LoL, or Val—never played any of those lol).

Remember to compete with your **friends** and **classmates**, like who has the higher **point** count on DMOJ :0 (more problems you solve, and the harder the problems are, give you higher points, some contests are rated, giving you elo/rating too).

You can also make and compete with online friends too, by joining the DMOJ **discord** server. <https://dmoj.ca/about/discord/>





Practice (wow, finally doing something)

This is your homework, and I recommend doing it if you are interested in competitive programming or just want to improve your problem solving skills. Aim for solving them correctly or at least attempting to.

Implementation Practice

Junior (beginner friendly)

- 1) Hello, World! – <https://dmoj.ca/problem/helloworld>
- 2) A Plus B – <https://dmoj.ca/problem/aplusb>
- 3) CCC '14 J2 – Vote Count – <https://dmoj.ca/problem/ccc14j2>

Senior (challenging) (if junior is too easy, or you want more practice)

- 1) CCC '14 J1 – Triangle Times – <https://dmoj.ca/problem/ccc14j1>
- 2) CCC '23 J3 – Special Event – <https://dmoj.ca/problem/ccc23j3>
- 3) UACC 1 P2 – Puzzling Parks – <https://dmoj.ca/problem/uacc1p2>

Get used to DMOJ and the problem solving format. Problems in the homework will always be listed from easiest to hardest (my opinion though, not absolute difficulty). **Remember, output has to be in the exact same format required.** Also, **we will always take up the homework problems** (unless no one wants the solution for that problem), **during the next competitive programming presentation.**

Common Errors

<https://dmoj.ca/about/codes/>

AC: Your program passed testing!
You solved the problem correctly.

WA: Your program did not crash
while executing, but the output it
produced was wrong.

TLE: Your program went above the
problem's time limit.

MLE: Your program went above the
problem's memory limit.

Other errors:

Probably just means
your program **crashed**.

Read the **error message**
for more information.

Sometimes you might
get **presentation error**,
meaning the way you
are outputting is wrong.
(eg. you have trailing
whitespace).

Execution Results

✓✓✗✓✓✓✓

Test case #1: AC [0.136s,31.80 MB] (50/50)
Test case #2: AC [0.136s,31.80 MB] (50/50)
Test case #3: WA [0.147s,31.80 MB] (0/50)
Test case #4: AC [0.136s,31.85 MB] (50/50)
Test case #5: AC [0.135s,31.73 MB] (50/50)
Test case #6: AC [0.135s,31.87 MB] (50/50)
Test case #7: AC [0.135s,32.05 MB] (50/50)

Resources: 0.960s, 32.05 MB
Final score: 300/350 (0.0/3 points)

Execution Results

✓✓✓✓✓✓✓✓✗✗✗

Test case #1: AC [0.004s,3.52 MB] (1/1)
Test case #2: AC [0.003s,3.46 MB] (1/1)
Test case #3: AC [0.003s,3.50 MB] (1/1)
Test case #4: AC [0.003s,3.34 MB] (1/1)
Test case #5: AC [0.003s,3.51 MB] (1/1)
Test case #6: AC [0.003s,3.48 MB] (1/1)
Test case #7: AC [0.048s,3.26 MB] (2/2)
Test case #8: TLE [>0.500s,2.96 MB] (0/2)
Test case #9: TLE [>0.500s,3.03 MB] (0/3)
Test case #10: TLE [>0.500s,3.02 MB] (0/2)

Resources: ---, 3.52 MB
Final score: 8/15 (5.333/10 points)

Thanks!

Do you have any questions?

Jason Sun – 350702866@gapps.yrdsb.ca

Puneet Bhat – 349068395@gapps.yrdsb.ca

CREDITS: This presentation template was created by [Slidesgo](#),
including icons by [Flaticon](#) and infographics & images by [Freepik](#)