ASC Tutor Tips for CIS/COMP students

UNIVERSITY OF THE FRASER VALLEY
ACADEMIC SUCCESS CENTRE

Available online at: https://ufv.ca/asc/student-resources/

Tips for CIS/COMP Student Success at UFV

1) **Take notes in class:**
Make sure to take notes during class lectures. Your instructor may mention something that is not a part of the lecture slides. Note-taking is one of the most effective ways to learn, remember and understand a subject!

2) **Ask for help:**
It is okay if you don't understand all of the concepts right away. Don't hesitate to seek help from fellow students and lab monitors. Also, you can ask your instructor for help during their office hours or send them an email outside of class time. Remember—the ASC is a great resource if you need help and guidance!

3) **Form a study group:**
Studying in a group can be very productive and enjoyable. Regular meetings help solidify concepts and can improve grades! Use this time to review the material, go over assignments and discuss any other problems you experience in class. Keep in mind—a study group session is not a class, so have fun!

4) **Pay extra attention in foundational courses:**
Courses like COMP 125, COMP 150, COMP 155 and COMP 251 teach you the main concepts that will accompany you throughout your CS/CIS career. These "building blocks" equip you with skills that allow you to develop complex programs.

5) **Focus on the logic not the syntax:**
It is pretty common to forget the syntax of a specific programming language, but that's okay; focus on logic building and not cramming the syntax. Understanding the functioning and the logic behind the line of code is imperative. Try some logic building exercises and make sure you are well versed with the concept; flow charts offer a visual way to map out the logic behind the business process and code execution.

6) **Use standard programming practices:**
Make proper use of comments and make sure to divide your code into appropriate sections. Using comments to explain your code leads to better understanding and makes it easier to collaborate with other programmers. Follow common coding practices and always use proper indentation to increase the readability of the code.

7) Keep project requirements in mind:

When writing software, keep the purpose and requirements in mind. What is the problem you are trying to solve? What should be the input and expected output? Focus on the project's objective and the professor's expectations while developing a program. Going beyond the project scope might help, but don't overanalyze.

8) Include diagrams in your project reports:

Diagrams are a great way of visualizing processes and systems. In CS/CIS, you will be using UML, which stands for Unified Modelling Language, to model the design of a system. Think of it as a blueprint for your work! Some helpful UML diagrams include class, sequence and 'use case diagrams' (representations of a user's interaction with the system).

9) Don't underestimate the importance of math and stats:

Math and stats may not seem necessary while writing simple code, but they are essential when dealing with algorithm analysis and complex systems. Some crucial fields include discrete mathematics (logic, combinatorics), number theory (cryptography and security) and linear algebra (graphics and machine learning).

10) Be Responsible

Being a CIS/CS student, you have the ability to develop very important, complex and massive systems. But remember—with power, comes great responsibility. Be careful not to leave your system unattended and vulnerable to data breach and other attacks. Follow the standard security principles to make your program secure. Always give credit wherever due, and be a responsible developer.

11) Academic Integrity in Computing Science:

Academic Integrity in CS/CIS is no different than in other disciplines! Make sure to cite your sources and give credit where credit is due. Citations not only make your work more credible but also point the reader to useful papers or other resources. Your code is your own intellectual property. Never share your assignments or code with another student or peer, as that can compromise your academic integrity. Some citation styles in CS/CIS include ACM, APA and IEEE. If in doubt, ask your instructor which one to use!

12) Be social:

You don't need to learn on your own—Join an online community! GitHub, Reddit and Stackoverflow, are great platforms to exchange knowledge and to collaborate on joint projects. Networking is a great way to meet new people and find opportunities. Organizations like IEEE and ACM provide professional support and guidance to students in various technologies through seminars, national and international level competitions, webinars etc. Students can benefit by becoming a member and networking with other members locally or abroad. Remember to keep your LinkedIn profile up to date!

13) Keep yourself updated and BE CURIOUS:

CS/CIS are fast-moving fields with upcoming and new technologies emerging on a daily basis. Frequently reading relevant blogs and news about computer science allows you to learn about current industry topics. Embrace change, explore new ideas and engage in research.

14) Make the best out of your free time:

Semester breaks can be the perfect time to explore new fields and learn new skills. Have you always wanted to create your own video game, make an app or build a spam detector? Take advantage of your time-off and take an online course on Coursera, DATAcamp or Udemy or check the resources available on Youtube such as the freeCodeCamp.org channel.

15) Start working on your own projects:

It may not be possible to take on challenging topics for course projects, but you can also work on new and interesting projects during summer or winter break. Always keep improvising with changing technology. Follow your gut and find out-of-the-box solutions or ideas to problems and be innovative. Don't forget to add your projects to your resume/portfolio.

16) Gain work experience:

Don't wait until you graduate to find work experience. School is the best time to polish your skills and build your resume. Enroll for CO-OP and summer internships to acquire some practical industry experience. Visit the Career Fair at UFV for opportunities.

Additional resources:

1. https://www.geeksforgeeks.org A portal of articles, questions and tutorials
2. https://code.tutsplus.com/tutorials/top-15-best-practices-for-writing-super-readable-code--net-8118: Basic programming practices
3. W3schools.com: A great for basics
4. https://www.sanfoundry.com: Online training programs

Associations:

IEEE - Institute of Electrical and Electronics Engineers
(This is the largest professional technological organization—Student memberships are available)
ACM – Association for Machine Computing

Remember:

For understanding concepts of computer science, the best practice to follow is to implement the code! Through implementation the concept will become clearer for you to understand.

This resource was developed by ASC tutors Arfhad, Hieu, Uliyana, Vaibhav and Rajdeep, April 2020.

Follow us on social media:
@UFVasc

CONTACT: asc@ufv.ca www.ufv.ca/asc