

Branching Exercise #1

- The program will prompt for a course name and number "Course name and number (e.g. CPSC 203): "
- When the course is CPSC 203
 - The following message will appear "Non-majors: Solving problems with computers mostly by using pre-written applications"
- When the course is CPSC 217
 - The following message will appear "Non-majors: Procedural (functions) programming"
 - Next the program will prompt again for a course name and number "Course name and number (e.g. CPSC 203): "
 - When the course is CPSC 219
 - The following message will appear "Non-majors: Object-Oriented programming"
 - Next the program will prompt again for a course name and number "Course name and number (e.g. CPSC 203): "
 - When the course is MATH 271
 - The following message will appear "Completed all 1st year course requirements, can take many 2nd year courses such as CPSC 331"
- When the course is CPSC 231
 - The following message will appear "Majors course: Procedural (functions) programming"
 - Next the program will prompt again for a course name and number "Course name and number (e.g. CPSC 203): "
 - When the course is CPSC 233
 - The following message will appear "Majors course: Object-Oriented programming"
 - Next the program will prompt again for a course name and number "Course name and number (e.g. CPSC 203): "
 - When the course is MATH 271
 - The following message will appear "Completed all 1st year course requirements, can take many 2nd year courses such as CPSC 331"
- When the initial course entered is anything other than: CPSC 203, 217 or 231 then the program will display the message "OK"

JT's note:

- I will go over a solution in lecture.
- It may be possible to implement solutions that are different from the one presented.
- To determine if your alternative solution is correct don't just bring it to the course instructor or one of the TA's, this is yet another opportunity for you get more practice, this time via hand tracing your implementation.
- If things look correct with the hand trace then consider what reasonable set of tests are needed check the different cases for this program.
- Of course if you find bugs during the hand trace you should fix them.
- Run your solution using these test cases.
- Again if you find unexpected (incorrect) results for a given set of inputs then you should hand trace it again and try to figure out where the bugs are and try to fix them.
- We are happy to help but the real value of this exercise is to try it yourself:
 - Try to implement a solution
 - Hand trace the solution
 - Test the solution
 - Try to fix any bugs
 - If you are stuck after trying as many of these steps as possible then that's where you can seek help.
 - If you are stuck on the first step after giving it a real attempt then watch part of the lecture video.
 - I'll likely go over the exercise after I have completed branching but before starting loops.
But try to watch as little of the video as possible.
 - That will allow you to get started but hopefully leave you with enough the remaining problem so you can still get something out of the exercise.
 - (Just watching someone else going over yet another problem won't be nearly as valuable as you trying to work through the problem yourself – remember the workout metaphor used under the '4 tips for success'.