Aida Sharif Rohani

1. Describe how you would integrate CodeQL code quality queries into your project development process.
   (a few sentences)

First I generate a code database from all our project codes and will write queries to find bugs, weaknesses and check security in my code. For example I can find all the functions that are made but never reached by a callee. In this case I can remove such functions and make my code cleaner.In terms of security I can write queries to check if a user or application can have access to the vulnerable functions in my code. Last by not least I can check the syntax format of all functions, constants, files, check if all comments are inline, if line sizes are 140 characters, and fix them if they don't follow our predefined style standards.

2. Is it possible to write a CodeQL query that finds functions that do not execute more than 10 times?
   (Yes/No, explain your answer in a few sentences)

   No, in case the function calls are dynamic, static analysis cannot find the callee and we can't find out how many times such functions are executed. As it was explained in lecture the best we can do is to report a range of calling times (0 to 20 times for example). An example is that when a function is only executed based on a condition given by a user input which could be only known at run-time. However, when all the identifiers of a callee are statically identifiable, and for example the sources of callee are not nested in a conditional dynamic loops, by using the static call graph we can find out if those static functions are called more than 10 times or not. Another *example is when we don’t find any callee sources for a function in the code database we know for sure that function will not execute at all.*

3. Is it possible to write a CodeQL query that finds all loops that will execute more than 100 times?
   (Yes/No, explain your answer in a few sentences)

   No, when the loops are dynamic (it means the loop relies on a variable passed into the function or maybe it relies on input from the user or etc.), the number of iterations is only determined at run-time. By static analysis of dynamic loops we can only find the worst case execution time and not the exact number of executions. However, when a loop is statically identifiable and its execution is not dependent on dynamic conditions at run-time (like while a condition given by a user, or a loop inside a function which its call is dynamic) we can count the number of iterations and see if it was iterated more than 100 times. For example when in the static main function node we have the following loop:

   ```
   for (int i=0, i<200,i++){
   print(1)
   }
   ```

   In such static condition we know for sure it runs more than 100 times.

4. Is it possible to write a CodeQL query that finds all variables that are constants, but whose names are not in all caps?
   (Yes/No, explain your answer in a few sentences)

   Yes we can identify the constant expressions and we can also check if they are upper or Lowe case, in CodeQL we can find constants by constExpr and we can check if all letters are uppercase or not by toString().isUpperCase.