CMPT 470 Final

1. Part 1: Introduction to TXL (this part will be asked by different types of question including writing TXL programs)
   1. What is TXL good for? Describe with examples.
   2. Couple small questions more on TXL. Highlighted parts of the TXL manual.
      1. Grammars
      2. Rules
   3. Write TXL grammar for a given code. In lectures.
      1. Asked to redefine the grammar for parsing a new but related input.
      2. Calculator program
   4. Writing a couple of small TXL programs that also exercise the TXL rules such as the
      1. calculator program
      2. or the number summation problem
   5. Grammar for the command language
      1. If else
      2. Case
      3. Switch /case
      4. Transformation
         1. Transfer one to another
      5. Few missing lines in the replacement part of one of the major rules
   6. Need to read
      1. Introductory
      2. TXL manual
      3. TXL cookbook
      4. Example
2. Vaclav Software Engineering Books
   1. Must know
      1. Why do we care about change? Know the basic of software change including the software change process model
      2. Concept location, need to know this detail including the rows of concept tringle
      3. Given a change request and then will ask you to find out the significant concept out of it, justify
      4. Know the different methods of conception location, including their importance and roles, strengths and weaknesses
      5. Decency graph and then ask to find the significant concept by suitable conception by suitable concept location methodology
      6. Ask about impact analysis. Test knowledge of different impact analysis method in lecture, strengths and weakness.
      7. Example scenarios and then will say apply method x for impact analysis or concept location
         1. E.g. test knowledge on those methods for a particular example scenario
      8. Provide some code sell example, ask to identify possible refactoring candidates.
      9. Understand the situations of different code smell
      10. Lots of small question form software change
   2. Need to read
      1. Software change
         1. Intro
         2. Concept location
         3. Impact analysis
         4. Actualization
         5. Refactoring
         6. Verification
         7. Conclusion
      2. Change-based processes
         1. Change-based software process Models
         2. Software Engineering Life Span Models
         3. Intro SE
         4. Case Studies on Software change
         5. Case Studies on software process
3. Other advanced topics related S E
   1. Liskov Substitute principle
      1. Know what it is
      2. What are issues
      3. Someone can violate this
      4. how to solve the issue
   2. Open-close principle
      1. Object-Oriented design principle
      2. Know the details
   3. Other
      1. Actor Model
      2. Architectural Styles
      3. Big Data Analytics and Frameworks
      4. Block Chaim
      5. Cloud Computing
      6. Deep learning
      7. Machine Learning
      8. Design patterns and Anti-Patterns
      9. SOLID principles
      10. Webapp development