ICT167 ANS8

Java Exceptions:

- When there is a violation of the sematic constraints of Java, the Java virtual machine signals this error to the program as an exception
- EG- dividing by zero, name is too long for spacing allocated for it on the screen, index position in string is out of bounds

- ALWAYS generate a string/message for the user to indicate the exception caught

Java exception process:

- Exceptions are *thrown* by a program. Called throwing the exception
    - Either Java or code indicates when something unusual happens- involves creating exception object
- Exception may be *caught*
- Exceptions are then *handled* by another part of the (same) program. Called handling an exception
    - Responding to an exception by executing part of the program written for the exception

- Thus the program is separated into-
    - Normal execution flow
    - Exception execution flow
- EG- We writing report on Microsoft (normal execution) then error comes up like file can't be saved (exception) therefore we need to handle it via exception execution flow

Try block:

- Testing whether there any exceptions. A try block is followed by one or more catch blocks

Throw block:

- Essentially passes/throws a message CUSTOM (exception) to a catch block. Will go straight to catch block and execute the code down in the catch block.
- If there is no throw block in the try block than the CUSTOM catch block is obviously skipped

Catch block:

- Take in parameter to identify the type of exception to perform

- Some catch blocks do not require throw blocks. These catch block are in pre-defined catch blocks not custom catch blocks. EG
    - Catch(IOException E) {   }
    - Catch(ClassNotFoundException) {   }

- Catch(FileNotFoundException) { }
- Catch(IndexOutOfBoundsException) { }
- Catch(NumberFormatException) { }

(44Mins)