COMP1531

Week 7 Tutorial!

Housekeeping

- Project deliverable 2:
  - UML class diagram - feedback this lab
  - back-end implementation - submit next Sunday
  - demo - week 08 lab
- Lab 07 - due week 08 Sunday

Equivalence Class

Eq. Class - Student Grade

- Given a mark - return a grade e.g. PS, HD

Answer

- FL: 0 - 49
- PS: 50 - 64
- CR: 65 - 74
- DN: 75 - 84
- HD: 85 - 100
- Invalid mark: mark < 0 or mark > 100
- Invalid value: not a number

Coding example

Requirement: “the length of the user-name >= 1 and <= 25 and not contain a space”

Design a function validate_user_name() that takes in as input a user-name and performs the above validation.

Eq. Classes

Field	Description
Valid equivalence class	
User Name	Length >=1 and <=25 and not containing a space
InValid equivalence class	
User Name	Empty
User Name	Length > 25 (No spaces)
User name	Length >=1 and <= 25 and containing a space

Test cases

Testcase	Input	Expected Output
#1	“nyancat”	True
#2	“”	UserInputError
#3	“aaaaaaaaaaaaaaaaaaaaaaaaaaa.....”	UserInputError
#4	“nya cat”	UserInputError

Flask

Flask (i)

- function `index()` is registered as the handler for the application's root URL.
- The return value of the function is called the response which is what is received by the client (e.g. web browser)

- 127.0.0.1 is an IP Address for Local Host
- The interactive debugger allows you to execute code directly in your browser at any specific point in the stack trace.

Flask (ii)

- The methods=["GET", "POST"] tells flask that this function will now have support for “POST” requests. Remember that we used “POST” request in our form in index.html.

Jinja

- Jinja2 is a friendly templating language for Python used to create HTML, XML or any other markup formats.
- A Jinja2 template contains markup code mixed with variables which are replaced by the values which are passed in when the template is rendered.
- To render a template you can:
  - use the render_template() method.
  - provide the name of the template and the variables you want to pass to the template engine as keyword arguments.
  - `e.g., render_template("hello.html", name=name, id=zID, desc=description)).`
- The template engine will replace the {{var}} variables in hello.html with the value of the variables