CSE/ISE 337 - Scripting Languages
Spring, 2021
Grading Guide for Final Exam

General Instructions:
- -Each short answer question is worth 5 points.
- -For each short answer question, I have listed several points that I think that all correct answers must contain. I have also included a complete sample answer for comparison.
- -Please use your discretion in interpreting the student's answers.
- -If the set of listed deductions does not sum to 5, the remaining points are given for free. Unless the student has left the question blank. Empty answers get 0 points.
- -Favor generosity over strictness
- -If you are not sure, ask me.
- -Each question is on a separate page.

Question 61: Qiaomu Miao
Question 62: Shilin Hu
Question 63: Chenlu Wang
Question 64: Omkar Bhatt
Question 65: Piyush Kidambi
Question 66: Akshay Nehe
Question 67: Rajesh Prabhakar
Question 68: Me
Question 69: Me
Question 70: Me

61. Python: Given a list of integers, lst*, write a list comprehension that builds a list of integers from* lst *that are the even squares of the elements of* lst*.*

Grader: Qiaomu Miao

Evaluation:

1.0 pts. for using a list comprehension
1.0 pts. correct syntax for looping over *lst*
1.0 pts. correct test for even square
-(note x^2 is even iff. x is even)
1.0 pts. correct calculation of new list element
1.0 pts. for free

My answer for comparison:

```
[(x*x) for x in lst if ((x*x) % 2) == 0]
```

62. Python: Given the following Python function:

```
def f(a, b):
  if a == b:
    return a
  elif a > b:
    return f(b, a - b)
  else:
    return f(a, b - a)
```

i. Explain what this function does. What problem does it solve?

ii. For a = *10,* b = *4, give the result, state how many recursive calls will occur, and explain why.*

iii. For a = *5,* b = *3, give the result, state how many recursive calls will occur, and explain why.*

Grader: Shilin Hu

Evaluation:
- 2.0 pts. for explaining what the function does
  - -Only 1.5 points if they explain the steps without explaining what the computation really does
  - -Partial credit here
- 1.0 pts. each for cases (ii) and (iii)
  - 0.5 pts. for stating the right result
  - 0.5 pts. for stating the right number of recursive calls.
  - -You can ignore the "explain why"
- 1.0 pts. for free

My answer for comparison:

i. This function calculates the greatest common divisor (GCD) of the two inputs.

ii. The result is 2 and it requires 3 recursive calls. Inputs to recursive calls are (4, 6), (4, 2),

(2, 2). The last recursive call hits the base case and returns 2.

iii. The result is 1 and it requires 3 recursive calls to shows these two are relatively prime. Inputs to recursive calls are (3, 2), (2, 1), (1, 1). The last recursive call hits the base case and returns 1.

63. Regular Expression: Given the following regular expression and string:

```
r"((\w+://)((\w+)[.]((\w+)[.](\w+))))"
s = "http://www.apple.com"
```

Give the substrings matched by each group in order by group number.

Grader: Chenlu Wang

Evaluation:
  0.7 pts. off per incorrect group

My answer for comparison:

```
G1 = "http://www.apple.com"
G2 = "http://"
G3 = "www.apple.com"
G4 = "www"
G5 = "apple.com"
G6 = "apple"
G7 = "com"
```

64. UNIX Shell scripting: What does the following command do?

```
head -n 1 *.* | grep -E "^.*[Pp]ython.*$" | cat > out.txt
```

Grader: Omkar Bhatt

Evaluation:

If they give a single, unified answer:

5.0 pts. if they give what amounts to the TLDR answer below

If they break down the command then:

1.0 pts. gets first line of every file

1.0 pts. for grep filters out lines that don't contain "Python" or "python"

1.0 pts. for cat writes lines from grep out

1.0 pts. for output of cat redirected to out.txt

1.0 pts. for free

My answer for comparison:

TLDR: This command will copy the first line of every file in the directory such that the first line contains the string Python or python. All those first lines are written to out.txt.

Stepwise: the first command reads just the first line from every file in the directory. That collection of first lines if fed to grep which filters out all the first lines that do not contain a match of the specified regular expression pattern, Python or python surrounded by anything but newline. The filtered collection of first lines is then piped to cat where it is redirected to the file out.txt.

65. GUI Scripting: Explain how we used Event-Driven (or Asynchronous) programming to implement GUIs using Tkinter?

Grader: Piyush Kidambi

Evaluation:

1.0 pts. GUI waits for arbitrary user interaction
1.0 pts. GUI elements listens for user actions
1.0 pts. user interaction with GUI elements trigger callback functions
1.0 pts. callback functions execute actions in response to user interaction with GUI elements
1.0 pts. for free

My answer for comparison:

Event-Driven programming relies on the occurrence of unplanned arbitrary interaction with the user to drive the application. The GUI is setup to listen for user interaction with the GUI of various kinds (button presses, mouse clicks, slider repositioning, scrolling, item selection, etc.). User interactions are recorded as events. These events then trigger waiting callback functions to execute an action in response to the user's interaction with a GUI element.

66. Web scripting: Describe the procedure for defining a new model used to represent application data in the database for a Django web application.

Grader: Akshay Nehe

Evaluation:

1.0 pts. for models defined as Python class
1.0 pts. for classes extend models.Model
-full name/path for Django class not necessary
1.0 pts. for definitions use predefined data types
1.0 pts. for using makemigrations and migrate to add/update model to database
1.0 pts. for free

My answer for comparison:

A new model is defined as a Python class in the models.py file in the directory for the owning app within the Django project. This class will extend the Django.db.models.Model class. This Python class will contain a definition of instance variables, often using pre-defined Django data types. Once the class definition is done, a corresponding representation (a table) is built in the database using the makemigrations and migrate Django commands.

67. Prototyping: State the 3 criteria I claimed are necessary for a prototyping of an algorithm and then explain how using Scripting Languages can satisfy those criteria.

Grader: Rajesh Prabhakar

Evaluation:

1.0 pts. for efficient implementation and change
1.0 pts. for easy to read and understand
1.0 pts. for actually runs
1.0 pts. for explaining how scripting languages satisfy these criteria
1.0 pts for free

My answer for comparison:

A prototype for an algorithm must be efficient to write and update (in all senses, we can't waste all our development resources on the prototype), it must be easy to read and understand (so it can be studied by the developers), and it must be executable (we want to be able to run the prototype).

Scripting languages are designed for this kind of high-level work. Scripting languages allow for quick and efficient implementation and update. Scripting languages use abstractions that make programs easy to read and understand. Finally, prototype implementations of algorithms in scripting languages will run.

68. Distributed Algorithms: State the condition a process must satisfy in order to enter the critical section in Lamport's Distributed Mutual Exclusion algorithm.

Grader: Me

My answer for comparison:

In order for a process to enter the critical section, it must have sent out a request to all other nodes. Then it must wait for the following condition to be satisfied.

First, it must have received a reply from every other node and that reply must have a later timestamp than the processes request (they are replies to that specific request).

Second, it must check its queue to ascertain that its own request has the lowest timestamp of all the requests in its queue (using process ids to break ties).

Only if both conditions are met can the process enter the critical section.