|  |
| --- |
| **CS4720 Internet Programming**  **Department of Computer Science**  **Kennesaw State University**  **Test2**  **Summer 2021**  **Tuesday, July 27, 2021 (till 11:59pm)**  **Full Marks: 100**  **There are 13 Questions for a total of 100 points.**  **Answer ALL the questions** |

**NAME: KSU NetID:**

***General Instructions***

* First, ensure you have all the 9 pages of this exam booklet even before starting
* This exam is closed notes and closed books. No discussions are permitted
* Do not bring out your cell phone; don’t answer the phone; don’t read text messages
* You have 3 hours to complete the exam
* Write your answers clearly
* The size of the space given for each answer is sufficient
* Write no more than 3-4 lines for each of the short questions
* Even if your final answers are incorrect, you will get partial credit if intermediate steps are clearly shown to highlight thought process. This applies to program tracing questions as well.

**DISCLAIMER:** By submitting the completed pdf file, you are pledged that you did not consult any one during the Exam time and everything was closed books and closed notes.

|  |
| --- |
| Good Luck! |

1. **[2 pt ]** What is the difference between floating point (/) division and integer division (//) ?

|  |
| --- |
| // is apart of integer calculator with remainders and / will give you just the integer or number in front of the decimal place |

1. **[3 pt ]** Use a for loop to print values of the list [3, 2, 1, 0]

|  |
| --- |
| numbers = [3,2,1,0]  for number in numbers:  print(number) |

1. **[5 pt ]** This program is to do the following things.
   1. Let the user input the user id and password.
   2. Use the input id and password to connect to a database called courses.
   3. Output the number of students whose score is greater than 90. (hint: you may use the SQL command “select \* from grade where score>90”)

import **sqtile3**

i d=i nput ( ’ i d ’ )

password=input ( ’ password ’ )

db=sqlite3. connec t ( ‘NameOfYourDB’)

try :

-----------------------------

result= print ( result )

except :

print ( “Error ” )

|  |
| --- |
| import sqlite3  connection = sqlite3.connect('courses.sq')  cursor = connection.cursor()  select\_above\_90 = 'select \* from grade where score > 90'  id = input()  password=input()  try :  result = cursor.execute(select\_above\_90)  for row in result:  print(row)  except:  print ("Error") |

1. **[5 pt ]** Choose a number between 1 and 10 and assign it to the variable secret. Then, select another number between 1 and 10 and assign it to the variable guess. Next, write the conditional tests (if, else, and elif) to print the string ‘too low’ if guess is less than secret, ‘too high’ if greater than secret, and ‘just right’ if equal to secret.

|  |
| --- |
| import random  n = random.randint(1, 10)  guess = int(input("Enter an integer from 1 to 10: "))  while n != "guess":  print  if guess < n:  print ("guess is low")  guess = int(input("Enter an integer from 1 to 10: "))  elif guess > n:  print ("guess is high")  guess = int(input("Enter an integer from 1 to 10: "))  else:  print( "you guessed it!")  break |

1. **[5 pt ] List Comprehension**

Use lists of numbers and strings to represent elements in the real world with great variety.

* 1. Create a list called years\_list, starting with the year of your birth, and each year thereafter until the year of your fifth birthday. For example, if you were born in 2000, the list would be years\_list = [2000, 2001, 2002, 2003, 2004, 2005]
  2. In which year in years\_list was your third birthday? Remember, you were 0 years of age for your first year.
  3. In which year in years\_list were you the oldest?

|  |
| --- |
| years\_list = [1997, 1998, 1999, 2000, 2001, 2002]  years\_list[2] //third year  years\_list[-1] //oldest |

6. **[10 pt ] Dictionary**

**6.1** Make an English-to\_french dictionary called **e2f** and print it. Here are your starter words: dog is chien, cat is chat, and walrus is morse.

|  |
| --- |
| e2f = {'dog': 'chien', 'cat': 'chat', 'walrus': 'morse'} |

6.2 Using your three-word dictionary **e2f**, print the French word for walrus.

|  |
| --- |
| print(e2f['walrus']) |

7**. [10 pt ] Function**

Define a generator function called get\_odds() that return the odd numbers from range(10). Use a for loop to find and print the third value returned.

|  |
| --- |
| get\_odds = (num for num in range(10) if not num % 2 == 0)  count = 0  for num in get\_odds:  if count == 2:  print(num)  break  count += 1 |

8. **[10 pt ] Module**

**8.1** Create a file called *zoo.py*. In it, define a function called hours() that prints the string ‘Open 9-5 daily’. Then, use the interactive interpreter to import the zoo module and call its hours() function.

|  |
| --- |
| Zoo.py  def hours():  print('Open 9-5 daily')  anotherfile.py  import zoo  zoo.hours() |

8.2 Import the hours() function as info ad call it.

|  |
| --- |
| import as info zoo  info.hours() |

9. **[15 pt ] Database sqlite3**

9.1 Use the sqlite3 module to create a SQLite database called *books.db* and a table called books with these fields: title (text), author (text), and year (integer)

|  |
| --- |
| sqlite3 books.db  import sqlite3  connection = sqlite3.connect('books.db')  cursor = connection.cursor()  cursor.execute('''CREATE TABLE books(title TEXT, author TEXT, , year INT''')  select\_title\_alpha = 'select \* from books order by title'  select\_publication\_alpha = 'select \* from books order by author' |

9.2 Select and print title column from the book table in alphabetical order.

|  |
| --- |
| try :  result = cursor.execute(select\_title\_alpha)  for row in result:  print(row)  except:  print ("Error") |

9.3 Select and print all columns from the book table in order of publication.

|  |
| --- |
| try :  result = cursor.execute(select\_publication\_alpha)  for row in result:  print(row)  except:  print ("Error") |

10. **[5 pt ] CGI/WSGI**

A. What do you understand by CGI and WSGI?

|  |
| --- |
| WSGI Specific to python it allows python web apps to be long running processes and talk to the webclient. CGI its a general idea for any language. Allows the backend process to connect to the client through the webserver using request and responses. There is a new process for every request. |

B. Write a short note on HTTP GET vs POST method in support of HTML Form ACTION

|  |
| --- |
| HTTP GET Will allow you to get information from the server HTTP POST Will allow you to give the server information and the it will GET you some information if its need but it could just give the server the data and allow get a success or failure response code in the http header FORM ACTION Will allow you to do one of these methods when you submit a form online |

11**. [5 pt ] JSON**

Write a program that prompts for the user’s favourite number. Use json.dump() to store this number in a file. Write a separate program that reads in this value and prints the message, “I know your favourite number!. It’s -------”.

|  |
| --- |
| import json  number = int(input("What's your favourite number? "))  with open("numbers.json", 'w') as document:  json.dump(number,document)  with open("numbers.json") as document:  number\_ = json.load(document)  print("I know your favourite number! It's " + str(number\_)) |

12. **[10 pt ] Misc**

12.1 Write a short note on Web API and REST

|  |
| --- |
| Web API's can be called the backbone of the internet without these applications and the internet would be far less populated.  The nice thing about web API is it's integrated with common tasks a developer would normally do in web development. As such, it's not that hard to implement at all, and is very straightforward. Using the REST interfaces we won't go into much detail but we can look at how simple it is. There are three methods for communicating with the API's: POST, GET, and PUT.  the first one is simple to understand, so let's start with it.  POST is our traditional route. We send out a message with a body that defines what we want to see and it's optional parameters, you can use a keyword, like posts , to get a list of posts in that category. |

12.2 Share your experience how did you install and configured PyCharm, Django and Flask

|  |
| --- |
| To install Pycharm I went to the jetbrains website and download the DMG file and opened it on my computer and then moved it into my application folder the following installs of django and flask were much easier because pycharm had pre build applications using them |

13. **[15 pt ] Dijango**

1. What is Django and how do you use it?

|  |
| --- |
| Django is a web framework developers can use to write web applications. |

1. What is the role of Python using Django?

|  |
| --- |
| Python is the language that django is developed in |

1. Explain what is Model View Template (MVT) architectural pattern.

|  |
| --- |
| The model and the view are controlled by the python code written and the template is written in a templating language for django. The developer provides the Model, the view and the template then just maps it to a URL and Django serves it to the user. |

1. What are the features available in Django web framework?

|  |
| --- |
| 1. MultiLingual Support Framework Support Administration GUI Development Environment |

1. Explain the advantages of Django

|  |
| --- |
| MultiLingual Support- You can write your website and have it automatically translated to loads of languages through its internationalization system. Framework Support- suppoers ajax rss and chaching and other cool features that would be hard to implement yourself Administration GUI - comes with tools to build parts to manage site out of the box Development Environment - very lightweight development environment making it easy to use it for the whole stack |