CIS 30A Final Exam **Name** Aamir Khan

You can use any resources in the Canvas and your textbook but do not Google or use Chegg or any other internet resources. Do not discuss programming assignment questions with anyone including your classmates. If you have any questions about the programming assignment, contact your instructor. Do not discuss Midterm questions in Discussion Board.

Please submit your work to Canvas by 11:59 PM 12/11/2022. Be sure to submit your work as single pdf file along with signed signature page

NO LATE ASSIGNMENT WILL BE ACCEPTED.

"I affirm that I will not give or receive any unauthorized help on this exam, and that all work will be my own."

Signature ____________________

(2 points) Q1. What is the main difference between for loop and while loop in python?

For loop is used for definite iteration. while loops are used for indefinite iteration.

(2 points) Q2. What is argument and What is parameter in python functions. What is the difference between them? Arguments are the variables passed in a function call. Parameters tell us what kind of arguments we can have in the function definition.

(2 points) Q3. Explain what class is and what object is in python and give simple example explaining class and object. A class is a blueprint for objects.

An object is an instance created from the blueprint. You could have a Dog class and your object could be a golden retriever.

(2 points) Q4. Circle each reason we use functions:

[✓ a)] to reduce code duplication

b) to allow for the use of mathematics

[✓ c)] to make a program more modular

d) to reduce the number of variables.

(2 points) Q5. Circle the term that means, placing a decision inside of another decision

a) cohabitation b) spooning [✓ c) nesting] d) encapsulation

(2 points) Q6. For each statement about functions, tell true or false and briefly explain. Explanations are on the back.

a) A function must either return something or print something, or both. True

b) A function must have at least one parameter. False

c) A function cannot modify the value of any of its arguments. False

(12 points) Q7. Think of at least three kinds of your favorite pizza. Store these pizza names in a list, and then use a for loop to print the name of each pizza. Modify your for loop to print a sentence using the name of the pizza instead of printing just the name of the pizza. For each pizza you should have one line of output containing a simple statement like I like pepperoni pizza.

Add a line at the end of your program, outside the for loop, that states how much you like pizza. The output should consist of three or more lines about the kinds of pizza you like and then an additional sentence, such as I really love pizza! Use for loops.

On the back.

6)

A. A function can have both print statements and a return statement.

For ex:
```
def func():
    print("Hello")
    return 5
```

B. You can define functions with any number of parameters. You DONT need to have atleast 1 parameter.

C. Functions can modify the values of certain arguments like lists and dictionaries. For other objects, they are passed in by copy, so they can't be modified.

7)
```
pizzas = ['cheese', 'bbq', 'veggie']

for pizza in pizzas:
    print(pizza)

for pizza in pizzas:
    print(f"I really love {pizza} pizza!")
print("I really love pizza!")
```

Output:

```
pepperoni
hawaiian
veggie

I really love pepperoni pizza!
I really love hawaiian pizza!
I really love veggie pizza!

I really love pizza!
```

(12 points) Q8 Write a function that accepts a list of items a person wants on a sandwich. The function should have one parameter that collects as many items as the function call provides, and it should print a summary of the sandwich that is being ordered. Call the function three items, using a different number of arguments each time.

Output: On the back.

```
I'll make you a great sandwich:
  ...adding roast beef to your sandwich.
  ...adding cheddar cheese to your sandwich.
  ...adding lettuce to your sandwich.
  ...adding honey dijon to your sandwich.
Your sandwich is ready!

I'll make you a great sandwich:
  ...adding turkey to your sandwich.
  ...adding apple slices to your sandwich.
  ...adding honey mustard to your sandwich.
Your sandwich is ready!

I'll make you a great sandwich:
  ...adding peanut butter to your sandwich.
  ...adding strawberry jam to your sandwich.
Your sandwich is ready!
```

(14 points) Q9. Create a class call Employee and a subclass call Developer that inherits characteristics from Employee.

On the back of the last page.

In Employee class,

1) Define attribute location = "Riverside, CA"

2) Define __init__ that initializes name, email and role

3) Define get_info function that retrieves name, email and role using .format

8)

```
def make_sandwich (*items):
    print(" I'll make you a great sandwich:")
    for item in items:
        print(f" ...adding {item} to your sandwich.")
    print(" Your sandwich is ready!")
    print()
```

Driver code:

```
make_sandwich('roast beef','cheddar cheese','lettuce','honey dijon')
make_sandwich('turkey','apple slices','honey mustard')
make_sandwich('peanut butter','strawberry jam')
```

4) Define change_locale using any method we have learned that changes location

Class definitions on back.

In Developer sub class.

1) Define __init__ that initializes name, email, role and language

2) Use super().__init__ to initialize name, email and role and initialize language

3) Define get_info function that retrieves name, email and role using .format

---

1) Now, instantiate object call employee_1 for Mickey and employee_2 for Donald that passing following arguments to Developer.

"Mickey Mouse", "mmouse@disney.com", "Lead Character", "Python"

"Donald Duck", "dduck@disney.com", "Bad character", "FORTRAN"

2) Print employee_1 location

3) Change the location to Seattle WA using @classMethod

4) Print employee_1 location

5) Print employee_1 email

6) Print employee_1 name

7) Print employee_2 language

```
employee_1 = Developer ("Mickey Mouse", "mmouse @ disney.com",
                "Lead character", "Python")
employee_2 = Developer ("Donald Duck", "dduck @ disney.com",
                 "Bad character", "FORTRAN")

print(employee_1. location)
 employee_1. change_locale ("Seattle, WA")
 print (employee_1. location)
 print (employee_1. email )
 print (employee_1. name)
 print ( employee_2. language)
```

9)

```
class Employee:

    location = "Riverside, CA"

    def __init__(self, name, email, role):

        self.name = name

        self.email = email

        self.role = role

    def get_info(self):

        print("Name={} Email={} Role={}"
              .format(self.name, self.email, self.role))

    def change_locale(self, new_location):
        self.location = new_location


class Developer(Employee):

    def __init__(self, name, email, role, language):

        self.language = language

        super().__init__(name, email, role)

    def get_info(self):

        super().get_info()
```