100 DAYS OF CODE

Segun Folawiyo Ajidahun

CSC 202:

COMPUTER PROGRAMING II

125/21/1/0011

COMPUTER SCIENCE

WEEK 1 & 2

*#day 1*

*print("day 1-python print function")*

*print("hello world")*

*print("hello"+" "+"world")*

*input("what is your name?")*

*name = input("what is your name?")*

*length = len(name)*

*print(length)*

*a = input("a:")*

*b = input("b:")*

*c = a*

*a = b*

*b = c*

*print(f"a:{a}")*

*print(f"b:{b}")*

*print("weicome to band name generator.")*

*city = input("which city did you grow up in?\n")*

*pet = input("what is the name of a pet?\n")*

*print("your band name could be " + city +" " + pet)*

*#day 2*

*#PEDMAS*

*print(3\*3+3/3-3)*

*print(3\*(3+3)/3-3)*

*height = int(input("enter your height in m:"))*

*weight = int(input("enter your weight in kg:"))*

*bmi = weight/height\*\*2*

*print(bmi)*

*print("welcome to the tip calcultor")*

*bill=float(input("what was the total bill? $\n"))*

*tip=int(input("what percentage tip would you like to give? 10, 12, or 15?\n"))*

*people=int(input("how many people to split the bill?\n"))*

*tip\_as\_percent=tip/100*

*total\_tip\_amount=bill\*tip\_as\_percent*

*total\_bill=bill+total\_tip\_amount*

*bill\_per\_person=total\_bill/people*

*final\_amount=round(bill\_per\_person,2)*

*print(f"each person shhould pay: ${final\_amount}")*

DAY 3

print("Welcome to the roller coaster!")  
height = int(input("what is your height in cm?"))  
if height >= 120:  
 print("you can ride the roller coaster!")  
 age = int(input("what is your age?"))  
 if age<= 18:  
 bill = 12  
 print("youth tickets are $7")  
 elif age <= 12:  
 bill = 7  
 print("child tickets are $12")  
 else:  
 print("adult tickets are $23")  
 wants\_photo = input("Do you want a photo taken? Y or N.")  
 if wants\_photo == "Y":  
 bill += 3  
 print(f"your final bill is {bill}")  
  
else:  
 print("sorry, you have to grow taller before you can ride.")

*#love\_calculator*print('Welcome to love calculator!')  
name1 = input("what is your name?\n")  
name2 = input("what is their name?\n")  
combined\_string = name1 + name2  
lower\_case\_string = combined\_string.lower()  
t = lower\_case\_string.count("t")  
r = lower\_case\_string.count("r")  
u = lower\_case\_string.count("u")  
e = lower\_case\_string.count("e")  
true = t + r + u + e  
l = lower\_case\_string.count("l")  
o = lower\_case\_string.count("o")  
v = lower\_case\_string.count("v")  
e = lower\_case\_string.count("e")  
love = l + o + v + e  
love\_score = int(str(true) + str(love))  
  
if (love\_score < 10) or (love\_score > 90):  
 print(f"your love score is {love\_score}, love is not for you ")  
elif(love\_score >= 40) and (love\_score <= 50):  
 print(f"your score is {love\_score},love is for you.")  
else:  
 print(f"you score is {love\_score}")  
  
  
 *# Day 3 final project*print("welcome to Treasure Island.")  
print("your mission is to find the treasure.")  
choice1 = input('you\'re at a crossroad, where do you want to go? type"left" or "right".').lower()  
  
if choice1 == "left":  
 choice2 = input('you\'ve come to a lake.there is an island in the middle of the lake. type "wait" to wait for a boat. '  
 'type "swim" to swim across.').lower()  
if choice2 == "wait":  
 choice3 = input("you arrive at the island unharmed. there is a house with 3 doors. one red, one yellow and one blue."  
 "which colour do you chose.?\n").lower()  
if choice3 == "red":  
 print("its a room full of fire. game over.")  
if choice3 == "yellow":  
 print("you found the treasure! you win!")  
else:  
 print("you chose a door that does not exits. game over.")

*#Day 4  
#randomisation and python list*test\_seed = int(input("create a seed number:"))  
import random  
random\_side = random.randint(0, 1)  
if random\_side == 1:  
 print("heads")  
else:  
 print("tails")  
  
*#python list*user\_choice = input("what do you choose? type 0 for rock, 1 for paper or 2 for scissors\n")  
computer\_choice = random.randint(0 , 2)  
print(f"computer chose {computer\_choice}")  
if user\_choice == 0 and computer\_choice == 2:  
 print("user wins!")  
elif user\_choice == computer\_choice:  
 print("you win!")  
elif computer\_choice == user\_choice:  
 print("it is a draw !")  
else:  
 print("you have typed invalid no, you lose!")

*# Day 5  
#python loop*import random  
  
fruits = ["apple", "peach", "pear"]  
for fruit in fruits:  
 print(fruit)  
 print(fruit + "pie")  
 print(fruits)  
student\_scores = input("input a list of student scores")  
for n in range(0, len(student\_scores)):  
 print(student\_scores)  
student\_scores = [78, 65, 89, 55, 91, 64, 89]  
highest\_score = 0  
for score in student\_scores:  
 if score > highest\_score:  
 highest\_score = score  
print(f"the highest score in the class is: {highest\_score}")  
*#password generator*letters = ['a','d','f','u','p','p','g']  
numbers = ['0','1','2','3','4','5','3','9','5']  
symbols = ['!','@','#','%','#','@']  
print("welcome to the pypassword generator! ")  
nr\_letters = int(input("how many letters would you like in your password?\n"))  
nr\_symbols = int(input(f"how many symbols would you like?\n"))  
nr\_numbers = int(input(f"how many numbers would you like?\n"))  
for char in range(1, nr\_letters + 1):  
 password\_list = []  
 password\_list += random.choice(letters)  
for char in range (1, nr\_symbols + 1):  
 password\_list += random.choice(symbols)  
for char in range (1, nr\_numbers + 1):  
 password\_list += random.choice(numbers)  
 print(password\_list)

*#day 6  
#indentation  
# def turn\_right():  
# turn\_left()  
# turn\_left()  
# turn\_left()  
#  
# while not at\_goal():  
# if right\_is\_clear():  
# turn\_right()  
# move()  
# elif front\_is\_clear():  
# move()  
# else:  
# turn\_left()  
#  
#pratice*print("hello")  
num\_char = len("hello")  
print(num\_char)  
def my\_function():  
 print("hello")  
 print("bye")  
  
my\_function()

*#Day 7*word\_list = ["ardvark","baboon","camel"]  
import random  
chosen\_word = random.choice(word\_list)  
guess = input("guess a letter:").lower()  
for letter in chosen\_word:  
 if letter == guess:  
 print("right")  
 else:  
 print("wrong")  
print(f'pssst, the solution is {chosen\_word}.')  
display = []  
word\_lenght = len(chosen\_word)  
for \_ in range(word\_lenght):  
 display += "\_"  
  
guess = input("guess a letter:")  
for position in range (word\_lenght):  
 letter = chosen\_word[position]  
 print(f"curren position: {position}\n current letter: {letter}\n guessed letter: {guess}")  
 if letter == guess:  
 display[position] = letter  
 print(display)  
import random

*#Day 8*def greet():  
 print("hello")  
 print("how do you do jack akin?")  
 print("is not the weather nice today?")  
greet()  
*#function that allows for input*def greet\_with\_name(name):  
 print(f"hello{name}")  
 print(f"how do you do{name}?")  
greet\_with\_name("jack")  
 *#funtion with more than one input*def greet\_with(name, location):  
 print(f"hello {name}")  
 print(f"what is it like in {location}?")  
greet\_with("jack akin", "nowhere")  
greet\_with("nowhere", "jack akin")  
greet\_with(location="london", name="wale")  
  
  
  
*#final project*alphabet = ['a', 'b', 'c', 'd', 'e', 'f',  
 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n','o',  
 'p','q', 'r', 's', 't', 'u', 'v', 'w',  
 'x', 'y', 'z','a', 'b', 'c', 'd', 'e', 'f',  
 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',  
 'p','q', 'r', 's', 't', 'u', 'v', 'w',  
 'x', 'y', 'z']  
direction = input("type 'encode' to encrypt, type 'decode' to decrypt:\n")  
text = input("type your message:\n").lower()  
shift = int(input("type the shift number:\n"))  
*#create a funtion calle 'encrypt*def encrypt(plain\_text, shift\_amount):  
 cipher\_text = ""  
 for letter in plain\_text:  
 position = alphabet.index(letter)  
 new\_position = position + shift\_amount  
 new\_letter = alphabet[new\_position]  
 cipher\_text += new\_letter  
 print(f"the encoded text is {cipher\_text}")  
  
encrypt(plain\_text=text, shift\_amount=shift)  
*#create a funtion to 'decode'*def decrypt(cipher\_text, shift\_amount):  
 plain\_text = ""  
 for letter in cipher\_text:  
 position = alphabet.index(letter)  
 new\_position = position - shift\_amount  
 plain\_text += alphabet[new\_position]  
 print(f"he decoded text is {plain\_text}")  
if direction == "encoded":  
 encrypt(plain\_text=text, shift\_amount=shift)  
elif direction == "decode":  
 decrypt(cipher\_text=text, shift\_amount=shift)

*#DAY 9*programming\_dictionary = {  
 "Bug": "An error in a program that prevents the program from running as expected."  
 "function""A piece of code that you can easily call over and over again."  
}  
programming\_dictionary["loop"] = "The action of doing something over and over again."  
empty\_dictionary = {}  
*#programming\_dictionary = {}  
  
#print(programming\_dictionary)*programming\_dictionary["Bug"] = "A moth in you computer."  
*#print(programming\_dictionary)  
  
#loop through a dictionary*for key in programming\_dictionary:  
 print(key)  
 print(programming\_dictionary[key])  
student\_scores = {  
 "Harry": 81,  
 "Ron": 78,  
 "Hermione":99,  
 "Draco":74,  
 "Neville":62,  
}  
student\_grades = {}  
for student in student\_scores:  
 score = student\_scores[student]  
 if score > 90:  
 student\_grades[student] = "Outstanding"  
 elif score > 80:  
 student\_grades[student] = "Exceeds Expectations"  
 elif score > 70:  
 student\_grades[student] = "Acceptable"  
 else:  
 student\_grades[student] = "Fail"  
 print(student\_grades)  
*#Nesting*capitals = {  
 "France":"Paris",  
 "Germany":"Berlin",  
},  
*#nesting a list in a dictionary*travel\_log = {  
 "France": {"cities\_visited":["Paris", "Lille", "Dijon"],"total\_visits": 12},  
 "Germany": {"cities\_visited":["Berlin", "Hambury"],"total\_visits": 15},  
}  
travel\_log = [  
 {  
 "country": "France",  
 "cities\_visited":["Paris","Lille","Dijon"],  
 "total\_visits":9  
 },  
 {  
 "country": "Germany",  
 "cities\_visited": ["Berlin","Hambury","Stutt"],  
 "total\_visits": 5  
 },  
]  
def add\_new\_country(country\_visited, times\_visited, cities\_visited):  
 new\_country = {}  
 new\_country["country"] = country\_visited  
 new\_country["visits"] = times\_visited  
 new\_country["cities"] = cities\_visited  
 travel\_log.append(new\_country)  
add\_new\_country("nigeria",2,["moscom","america"])  
print(travel\_log)

*#Day 10  
#funtions with inputs*def format\_name(f\_name, l\_name):  
 if f\_name == "" or l\_name == "":  
 return "you did not provide valid inputs."  
 formated\_f\_name = f\_name.title()  
 formated\_l\_name = l\_name.title()  
 return f"Result:{formated\_f\_name} {formated\_l\_name}"  
print(format\_name(input("what is your first name?"),input("what is your last name")))  
  
def is\_leap(year):  
 if year %4 == 0:  
 if year % 100 == 0:  
 if year %400 == 0:  
 return True  
 else:  
 return False  
 else:  
 return True  
 else:  
 print("Not leap year")  
def days\_in\_month(year, month):  
 if month > 12 or month < 1:  
 return "Invalid"  
 month\_days = [31, 28,31,30.31,31,30]  
 if is\_leap(year) and month == 2:  
 return 29  
 return month\_days[month - 1]  
year = int(input("enter a year:"))  
month = int(input("enter a month:"))  
days = days\_in\_month(year, month)  
print(days)  
*#Calculator*def add(n1, n2):  
 return n1 + n2  
*#Subtration*def subtract(n1, n2):  
 return n1 - n2  
*#Multiplication*def multiply(n1, n2):  
 return n1 \* n2  
*#Divition*def divide(n1, n2):  
 return n1 / n2  
  
operations = {  
 "+": add,  
 "-": subtract,  
 "\*": multiply,  
 "/": divide  
  
}  
num1 = int(input("what is the first number?:"))  
  
for symbol in operations:  
 print(symbol)  
 operation\_symbol = input("pick an operation from the line above:")  
num2 = int(input("what is the second number?:"))  
calculation\_function = operations[operation\_symbol]  
answer = calculation\_function[num1, num2]  
print(f"{num1} {operation\_symbol} {num2} = {answer}")

*#Day 11 practices  
 # #Blackjack Capstone Project*import random  
from typing import List, Any  
def deal\_card():  
 cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10]  
 card = random.choice(cards)  
 return card  
def calculate\_score(cards):  
 user\_cards = []  
 computer\_cards = []  
 is\_game\_over = False  
for \_ in range(2):  
 computer\_cards = []  
 new\_card = deal\_card()  
 *#user\_cards.append(deal\_card())* computer\_cards.append(deal\_card())  
 user\_score = calculate\_score(deal\_card())  
 computer\_score = calculate\_score(computer\_cards)  
 print(f"your cards: {deal\_card()}, current score: {user\_score}")  
 print(f"computer first card: {computer\_cards[0]}")