Thursday Oct 3, 2024

In-class Handout

COSC 101C Intro to Computing I

Prof. Forrest Davis

Name:

Discuss and complete the following questions with the person nearest you. You **may** be asked to share your thoughts with the class.

1. Write a program that asks the user to enter two numbers and outputs whether or not the first number is evenly divisible by the second number. Note, you should use **main** and an additional function which handles the check of divisibility.

```
def divisibility(a: float, b: float) -> bool:
    if a % b == 0:
        isDivisible = True
    else:
        isDivisible = False
    return isDivisible

def main() -> None:
    a = float(input("First number: "))
    b = float(input("Second number: "))
    if divisibility(a, b):
        print(f"{a} is evenly divisible by {b}")
    else:
        print(f"{a} is not evenly divisible by {b}")

main()
```

2. Write a program using a main function and auxiliary functions, which asks for the number of hours an employee works each week and outputs whether they work full-time (40 or more hours), half-time (exactly 20 hours) or part-time (less than 40 hours, but not exactly 20 hours).

```
def checkHours(hours: float) -> str:
    if hours >= 40:
        classification = 'full-time'
    elif hours == 20:
        classification = 'half-time'
    else:
        classification = 'part-time'
    return classification

def main() -> None:
    hours = float(input("How many hours do you work a week? "))
    worker = checkHours(hours)
    print(f"You are employeed {worker}")

main()
```

3. Rewrite the following functions to use chained conditionals instead of nested conditionals

```
def nestedA() -> None:
    a = int(input("a="))
    if a >= 50:
        print("Large")
    else:
```

```
        if (a < 10):
            print ("Small")
        else:
            print("Medium")

def chainedA() -> None:
    a = int(input("a="))
    if a >= 50:
        print("Large")
    elif (a < 10):
        print ("Small")
    else:
        print("Medium")

def nestedB() ->  None:
    i = int(input("i="))
    j = int(input("j="))
    if i > 5:
        if j > 15:
            print("A")
    else:
        if j > 15:
            print("B")
    if i > 10:
        print("C")

def chainedB() -> None:
    i = int(input("i="))
    j = int(input("j="))
    if i > 5 and j > 15:
        print("A")
    elif j > 15:
        print("B")
    if i > 10:
        print("C")
```

4. Create test cases that generate all the output.

```
def mystery(students: int, desks: int, professors: int) -> None:
    if (students + professors) > desks:
        print( "There are not enough desks!")
    else:
        print("There are enough desks for everyone to have their own.")
        leftover = desks - (students + professors)
        if leftover >= students:
            print("There will be A LOT of empty desks.")
        elif leftover >= professors:
            print("There will be exactly",leftover,"empty desks.")
        elif leftover >= 1:
            print("There will be at least one empty desk.")
        elif leftover < 1:
            print("We are good")
        else:
            print("There will not be any empty desks." )
def main() -> None:
    mystery(28, 100, 5) # test 1
```

```
main()
```

```
There are enough desks for everyone to have their own.
There will be A LOT of empty desks.
```

```
mystery(28, 10, 5)
mystery(28, 50, 5)
mystery(28, 35, 5)
mystery(28, 33, 5)
# We can never get "There will not be any empty desks.". We should consider fixing t
```

```
There are not enough desks!
There are enough desks for everyone to have their own.
There will be exactly 17 empty desks.
There are enough desks for everyone to have their own.
There will be at least one empty desk.
There are enough desks for everyone to have their own.
We are good
There are not enough desks!
```