

Python for Computer Science and Data Science 1

(CSE 3651)

MINOR ASSIGNMENT-2: CONTROL STATEMENTS AND PROGRAM DEVELOPMENT

1. Write a program such that Python will ask you if it is raining or not. If your answer is "yes", Python will say "Carry an umbrella". If you type anything else, Python will say "Bye".

Ans: **Raining (simple if–else)**

```
rain = input("Is it raining? (yes/no): ")  
if rain.lower() == "yes":  
    print("Carry an umbrella")  
else:  
    print("Bye")
```

OUTPUT

Yes

Carry an umbrella

2. Write a program such that Python will ask you if it is raining or not. If your answer is "yes", Python will say "Carry an umbrella". If you say "no", Python will say "No need to carry an umbrella". If you type anything else, Python will say "Bye".

Ans: **Raining with more options**

```
rain = input("Is it raining? (yes/no): ")  
if rain.lower() == "yes":  
    print("Carry an umbrella")  
elif rain.lower() == "no":  
    print("No need to carry an umbrella")  
else:  
    print("Bye")
```

OUTPUT

Yes

Carry an umbrella

3. A.

Write a Python program to calculate a student's letter grade based on their numeric score using the following scale: A (90–100), B (80–89), C (70–79), D (60–69), and F (below 60). Additionally, provide a comment for each grade: "Excellent" for A, "Good" for B, "Average" for C, "Needs Improvement" for D, and "Failing" for F.

B. Write a program that reads the lengths of the three sides of a triangle from the user. Then display a message that states the triangles type.

Ans: A. Student Grade using elif

```
score = int(input("Enter your score (0-100): "))
```

```
if 90 <= score <= 100:
```

```
    grade, comment = "A", "Excellent"
```

```
elif 80 <= score <= 89:
```

```
    grade, comment = "B", "Good"
```

```
elif 70 <= score <= 79:
```

```
    grade, comment = "C", "Average"
```

```
elif 60 <= score <= 69:
```

```
grade, comment = "D", "Needs Improvement"  
else:  
    grade, comment = "F", "Failing"  
    print(f"Grade: {grade} - {comment}")
```

OUTPUT

95

Grade: A - Excellent

Answer: Triangle type:

```
a = float(input("Enter side 1: "))  
b = float(input("Enter side 2: "))  
c = float(input("Enter side 3: "))  
  
if a == b == c:  
    print("Equilateral triangle")  
elif a == b or b == c or a == c:  
    print("Isosceles triangle")  
else:  
    print("Scalene triangle")
```

OUTPUT:

1

2

3

Scalene triangle

- 4.** Write a program that takes an integer input from the user and tells the user whether the number is even or odd.

Ans: Even or Odd

```
num = int(input("Enter an integer: "))
```

```
if num % 2 == 0:
```

```
    print("Even")
```

```
else:
```

```
    print("Odd")
```

OUTPUT

5

Odd

5. Write a program that takes a year as input and determines whether it is a leap year or not.

A year is a leap year if:

- It is divisible by 4 → year \% 4 == 0
- But not divisible by 100 → year \% 100 != 0
(because years like 1900 are not leap years)

- OR it is divisible by 400 → year \% 400 == 0
(so 2000 is a leap year, even though it's divisible by 100).

So, the rule is:

- Divisible by 4 ✓**
- If divisible by 100, then must also be divisible by 400 ✓**

Ans:

```
year = int(input("Enter a year: "))
```

```
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
```

```
    print("Leap year")
```

```
else:  
    print("Not a leap year")
```

OUTPUT

2025

Not a leap year

6. Write a program that takes an integer input from the user and prints whether it is prime or not.

Ans: Prime check

```
num = int(input("Enter an integer: "))
```

```
if num < 2:
```

```
    print("Not prime")
```

```
else:
```

```
    is_prime = True
```

```
    for i in range(2, int(num**0.5) + 1):
```

```
        if num % i == 0:
```

```
            is_prime = False
```

```
            break
```

```
    if is_prime:
```

```
    print("Prime")  
else:  
    print("Not prime")
```

OUTPUT

7

Prime

7. In line with the previous question, write a program to compute the sum of all prime numbers below a user input number, e.g., Sum of all prime numbers less than 20 is 77.

Ans: Sum of Primes below N

```
n = int(input("Enter a number: "))  
prime_sum = 0  
for num in range(2, n):  
    is_prime = True  
    for i in range(2, int(num**0.5) + 1):  
        if num % i == 0:  
            is_prime = False  
            break
```

```
if is_prime:  
    prime_sum += num  
  
print(f"Sum of all primes less than {n} is  
{prime_sum}")
```

OUTPUT

55

Sum of all primes less than 55 is 381

8. Write a program that takes an integer input from the user. Use a while loop to continuously prompt for input until the user enters a positive number. If the final number is even, multiply it by 2 and if it is odd, square it. Display the results at the end.

Ans: Positive number with while loop

while True:

```
num = int(input("Enter a positive number: "))
```

```
if num > 0:
```

```
    break
```

```
if num % 2 == 0:  
    result = num * 2  
else:  
    result = num ** 2  
print("Result:", result)
```

OUTPUT

55

Result: 3025

9. Write a program to find the remainder when a user input number is divided by 5 using match case. If the user inputs a non-integer, Python should say "Invalid input" and stop.

Ans: Match case (remainder by 5)

try:

```
num = int(input("Enter a number: "))
```

```
match num % 5:
```

```
    case 0:
```

```
        print("Remainder is 0")
```

```
case 1:  
    print("Remainder is 1")
```

```
case 2:  
    print("Remainder is 2")
```

```
case 3:  
    print("Remainder is 3")
```

```
case 4:  
    print("Remainder is 4")
```

```
except ValueError:  
    print("Invalid input")
```

OUTPUT

97

Remainder is 2

- 10.** Write a program that takes a string as input and prints out all possible sub strings of the string using loops, e.g., if the input is "abc", the output should be "a", "ab", "abc", "b", "bc", "c".

Ans: All substring of a string

```
s = input("Enter a string: ")  
for i in range(len(s)):  
    for j in range(i + 1, len(s) + 1):  
        print(s[i:j])
```

OUTPUT

apple

a

ap

app

appl

apple

p

pp

ppl

ple

p

l

le

l
le
e

11. Write a program that functions as a simple calculator. It should continuously accept a pair of numbers and an operator (+,-, *, /) from the user and print the result. If the user types "exit", the program quits. Otherwise the program continues asking for a pair of input numbers.

Ans: Simple Calculator (loop until exit)

while True:

```
    user_input = input("Enter two numbers and an operator (or type 'exit'): ")
```

```
    if user_input.lower() == "exit":
```

```
        print("Calculator exiting...")
```

```
        break
```

```
    try:
```

```
        n1, op, n2 = user_input.split()
```

```
        n1, n2 = float(n1), float(n2)
```

```
if op == "+":  
    print("Result:", n1 + n2)  
  
elif op == "-":  
    print("Result:", n1 - n2)  
  
elif op == "*":  
    print("Result:", n1 * n2)  
  
elif op == "/":  
    if n2 != 0:  
        print("Result:", n1 / n2)  
    else:  
        print("Error: Division by zero")  
  
else:  
    print("Invalid operator")  
  
except:  
    print("Invalid input format. Try again (e.g., 5 + 3)")
```

OUTPUT:

Invalid input format. Try again (e.g., 5 + 3)

Result: 8.0

Invalid input format. Try again (e.g., 5 + 3)

Result: 3.0

Invalid input format. Try again (e.g., 5 + 3)

Result: 10.0

Invalid input format. Try again (e.g., 5 + 3)

Result: 3.0

Invalid input format. Try again (e.g., 5 + 3)

Invalid input format. Try again (e.g., 5 + 3)

Error: Division by zero

Invalid input format. Try again (e.g., 5 + 3)

Calculator exiting...

12. Write a program to find out the mean, median, and mode of 1, 2, 3, 2, 3, 4, 4, 4, 5, 4, 5, 6.

Ans: Mean , Median , Mode

import statistics

data = [1, 2, 3, 2, 3, 4, 4, 4, 5, 4, 5, 6]

mean = statistics.mean(data)

median = statistics.median(data)

```
mode = statistics.mode(data)
```

```
print("Mean:", mean)  
print("Median:", median)  
print("Mode:", mode)
```

OUTPUT

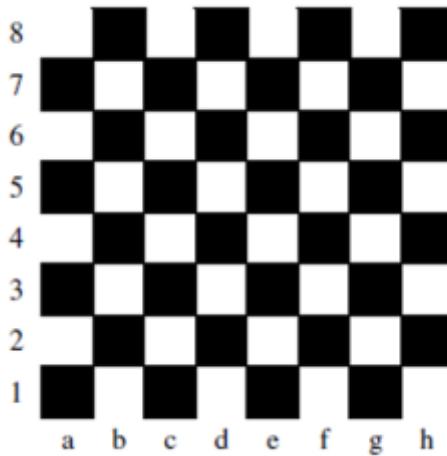
Mean: 3.583333333333335

Median: 4.0

Mode: 4

13. Positions on a chess board are identified by a letter and a number. The letter identifies the column, while the number identifies the row, as shown below:

Write a program that reads a position from the user and identify the proper color of the respective box.



Ans: We want a program that takes a chessboard position (like a1, c5, h8) and tells whether the square is black or white.

Logic:

- Columns (a–h) are letters → convert them to numbers (a=1, b=2, ..., h=8).
- Rows (1–8) are numbers directly.
- Rule:
 - If $(\text{column} + \text{row}) \% 2 == 0$ → **Black**
 - Else → **White**

```
position = input("Enter a chessboard position (e.g., a1,  
c5, h8): ").lower()
```

Extract column and row

```
col = position[0]
```

```
row = int(position[1])
```

Convert column letter to number (a=1, b=2, ..., h=8)

```
col_num = ord(col) - ord('a') + 1
```

Determine color

```
if (col_num + row) % 2 == 0:
```

```
    print(f"{position} is a Black square")
```

```
else:
```

```
    print(f"{position} is a White square")
```

OUTPUT:

a1

a1 is a Black square

14. Write a Python program that calculates the final cost of a hotel booking based on the room type (Standard: \$100/night, Deluxe: \$150/night, Suite: \$250/night), the length of stay (10% discount for >3nights, 20% discount for >7 nights), the season (20% increase during peak season, 15% decrease during the off-season), and whether the customer is a loyalty member (5% additional discount). The program should output the final booking cost after applying all relevant discounts and adjustments.

Ans: Hotel Booking Final Cost

```
room_type = input("Enter room type  
(Standard/Deluxe/Suite): ").lower()  
  
nights = int(input("Enter number of nights: "))  
  
season = input("Enter season (peak/off/normal):  
").lower()  
  
loyalty = input("Are you a loyalty member? (yes/no):  
").lower()
```

Room prices

```
prices = {"standard": 100, "deluxe": 150, "suite": 250}  
  
cost = prices[room_type] * nights
```

Discounts for stay length

if nights > 7:

cost *= 0.8 # **20% discount**

elif nights > 3:

cost *= 0.9 # **10% discount**

Season adjustments

if season == "peak":

cost *= 1.2 # **20% increase**

elif season == "off":

cost *= 0.85 # **15% decrease**

Loyalty discount

if loyalty == "yes":

cost *= 0.95

print("Final booking cost: \$", round(cost, 2))

OUTPUT:

Suite

7

peak

yes

Final booking cost: \$ 1795.5

15. Write a program to determine whether a given natural number is a perfect number. A natural number is said to be a perfect number if it is the sum of its divisors. For Example, 6 is a perfect number because $6 = 1+2+3$, but 15 is not a perfect number because $15 \neq 1+3+5$.

Ans: Perfect Number

```
n = int(input("Enter a natural number: "))
```

```
div_sum = sum(i for i in range(1, n) if n % i == 0)
```

```
if div_sum == n:
```

```
    print(n, "is a Perfect number")
```

```
else:
```

```
    print(n, "is not a Perfect number")
```

OUTPUT:

7

7 is not a Perfect number

16. Write a program that finds the sum of the n terms of the following series:

16. Write a program that finds the sum of the n terms of the following series:

a) $1 - x^2/2! + x^4/4! - x^6/6! + \dots + x^{2n}/(2n)!$

b) $1 + x/1! + x^2/2! + x^3/3! + \dots + x^n/n!$

c) $1-3+5-7+9-\dots$

a) Series Calculation

```
import math  
x = int(input("Enter x: "))  
n = int(input("Enter n: "))  
s = 1  
sign = -1
```

for i in range(1, n+1):

```
    s += sign * (x**2*i) / math.factorial(2*i)  
    sign *= -1
```

```
print("Sum of series (a):", s)
```

Output:

2

3

Sum of series (a): -0.422222222222223

b) $1 + x/1! + x^{}2/2! + \dots + x^{**}n/n!$**

`s = 1`

`for i in range(1, n+1):`

`s += (x**i) / math.factorial(i)`

`print("Sum of series (b):", s)`

OUTPUT:

Sum of series (b): 6.333333333333333

c) $1-3+5-7+9-\dots$

`s = 0`

`sign = 1`

`for i in range(1, 2*n, 2):`

`s += sign * i`

`sign *= -1`

`print("Sum of series (c):", s)`

OUTPUT:

Sum of series (c): 3

- 17. Write a python program that displays all the numbers from 100 to 1,000, ten per line, that are divisible by 5 or 6. Numbers are separated by exactly one space.**

Ans: Numbers 100–1000 divisible by 5 or 6 (10 per line)

```
count = 0
for i in range(100, 1001):
    if i % 5 == 0 or i % 6 == 0:
        print(i, end=" ")
        count += 1
    if count % 10 == 0:
        print()
```

OUTPUT:

**100 102 105 108 110 114 115 120 125 126
130 132 135 138 140 144 145 150 155 156
160 162 165 168 170 174 175 180 185 186
190 192 195 198 200 204 205 210 215 216
220 222 225 228 230 234 235 240 245 246
250 252 255 258 260 264 265 270 275 276
280 282 285 288 290 294 295 300 305 306
310 312 315 318 320 324 325 330 335 336
340 342 345 348 350 354 355 360 365 366
370 372 375 378 380 384 385 390 395 396
400 402 405 408 410 414 415 420 425 426
430 432 435 438 440 444 445 450 455 456
460 462 465 468 470 474 475 480 485 486
490 492 495 498 500 504 505 510 515 516
520 522 525 528 530 534 535 540 545 546
550 552 555 558 560 564 565 570 575 576
580 582 585 588 590 594 595 600 605 606
610 612 615 618 620 624 625 630 635 636**

640 642 645 648 650 654 655 660 665 666

670 672 675 678 680 684 685 690 695 696

700 702 705 708 710 714 715 720 725 726

730 732 735 738 740 744 745 750 755 756

760 762 765 768 770 774 775 780 785 786

790 792 795 798 800 804 805 810 815 816

820 822 825 828 830 834 835 840 845 846

...

880 882 885 888 890 894 895 900 905 906

910 912 915 918 920 924 925 930 935 936

940 942 945 948 950 954 955 960 965 966

970 972 975 978 980 984 985 990 995 996

1000

18. Write a Python program that prints all numbers from 1 to 100, except multiples of 7, using a for loop with continue.

Ans: Skip multiply of 7

```
for i in range(1, 101):
```

```
    if i % 7 == 0:
```

```
        continue
```

```
    print(i, end=" ")
```

OUTPUT:

```
1 2 3 4 5 6 8 9 10 11 12 13 15 16 17 18 19 20 22 23 24  
25 26 27 29 30 31 32 33 34 36 37 38 39 40 41 43 44 45  
46 47 48 50 51 52 53 54 55 57 58 59 60 61 62 64 65 66  
67 68 69 71 72 73 74 75 76 78 79 80 81 82 83 85 86 87  
88 89 90 92 93 94 95 96 97 99 100
```

19. Write a python program that accepts a positive integer n and reverses the order of its digits, e.g., 1234 becomes 4321.

Ans: Reverse Digit

```
n = int(input("Enter a positive integer: "))
```

```
rev = int(str(n)[::-1])
```

```
print("Reversed number:", rev)
```

OUTPUT:

9078225501

Reversed number: 1055228709

**20. Write a python program that reads an integer and displays all its smallest factors in increasing order,
e.g., if the input integer is 120, the output should be as follows: 2, 2, 2, 3, 5.**

Ans: Smallest factor in order

```
n = int(input("Enter an integer: "))
```

```
i = 2
```

```
while n > 1:
```

```
    if n % i == 0:
```

```
        print(i, end=" ")
```

```
        n //= i
```

```
    else:
```

```
        i += 1
```

OUTPUT:

10

25

21. Write a python program to determine whether or not a number n is a factorial number.

Ans: Factorial Number Check

```
n = int(input("Enter a number: "))
```

```
fact = 1
```

```
i = 1
```

```
while fact < n:
```

```
    i += 1
```

```
    fact *= i
```

```
if fact == n:
```

```
    print(n, "is a factorial number")
```

```
else:
```

```
    print(n, "is not a factorial number")
```

OUTPUT:

1

1 is a factorial number

22. Write a program that takes a number from the user and continuously sums its digits until the sum becomes a single-digit number.

Ans: Sum of digit until single digit

```
n = int(input("Enter a number: "))
```

```
while n >= 10:
```

```
    n = sum(int(d) for d in str(n))
```

```
print("Single digit sum is:", n)
```

OUTPUT:

125

Single digit sum is: 8

23. Write a program to simulate a simple ATM withdrawal system. The user can enter an amount they want to withdraw, and the program will provide the number of 100, 50, 20, and 10 denomination notes required to dispense that amount. The program should check if the requested amount is a multiple of 10 and if the ATM has enough cash.

Ans: Simple ATM Withdrawl

```
amount = int(input("Enter withdrawal amount: "))

atm_cash = 10000 # Example: ATM has $10,000

if amount % 10 != 0:

    print("Amount must be multiple of 10")

elif amount > atm_cash:

    print("ATM does not have enough cash")

else:

    notes = {}

    for denom in [100, 50, 20, 10]:

        notes[denom], amount = divmod(amount, denom)
```

print("Dispensed notes:")

for d, c in notes.items():

if c > 0:

print(f"\{d\}: \{c\}")

OUTPUT:

500

Dispensed notes:

100: 5

24. Write a program that reads an integer from the user and checks which digits (0-9) have appeared in the number. The program should print out the digits that have appeared, e.g. input=1234, output=ONETWOTHREEFOUR

Ans:

We want a program that:

- Reads an integer from the user.
- Checks which digits appear.
- Prints them in words (e.g., 1234 → ONETWOTHREEFOUR).

Dictionary to map digits to words

```
digit_words = {  
    "0": "ZERO",  
    "1": "ONE",  
    "2": "TWO",  
    "3": "THREE",  
    "4": "FOUR",  
    "5": "FIVE",  
    "6": "SIX",  
    "7": "SEVEN",  
}
```

```
"8": "EIGHT",
"9": "NINE"

}

num = input("Enter an integer: ")
output = ""

for digit in num:
    if digit in digit_words: # only digits
        output += digit_words[digit]

print("Output:", output)
```

OUTPUT:

1009

Output: ONEZEROZERONINE

25. Write Python programs using loops to print the following patterns:

** ** *** ****	*	***** *** ** *
**** *** ** *	**** *** ** *	* *** ***** ***** *** *
* * * * * * * *****	*	* * * * * * * * * * * * * * * *
0 1 2 3 4 5 0 1 2 3 4 0 1 2 3 0 1 2 0 1 0	1 3 3 5 5 5 7 7 7 7 9 9 9 9 9	1 2 1 3 2 1 4 3 2 1 5 4 3 2 1
1 2 4 3 6 9 4 8 12 16 5 10 15 20 25	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 [\	1 2 1 2 3 2 1 2 3 4 3 2 1 2 3 4 5 4 3 2 1 2 3 4 5

*

**

Pattern 1: Star Triangle (Left Column - Top)

for i in range(1, 6):

```
for j in range(i):  
    print("*", end="")  
  
    print()  
  
print("\n" + "="*30 + "\n")
```

Pattern 2: Reverse Star Triangle

**

*

Pattern 2: Reverse Star Triangle (Left Column - Bottom)

```
print("Pattern 2: Reverse Star Triangle")
```

```
for i in range(5, 0, -1):
```

```
    for j in range(i):
```

```
        print("*", end="")
```

```
        print()
```

```
    print("\n" + "="*30 + "\n")
```

Pattern 3: Hollow Square with Stars

```
* * * * *
```

```
* * *
```

```
* * *
```

```
* * *
```

```
* * * * *
```

Pattern 3: Star Diamond/Hollow Square (Left Column - Bottom)

```
print("Pattern 3: Hollow Square with Stars")
```

```
n = 5
```

```
for i in range(n):
```

```
    for j in range(n):
```

```
        if i == 0 or i == n-1 or j == 0 or j == n-1:
```

```
            print("*", end=" ")
```

```
        else:
```

```
            print(" ", end=" ")
```

```
    print()
```

```
print("\n" + "="*30 + "\n")
```

Pattern 4: Single Number Triangle

1

22

333

4444

55555

```
# Pattern 4: Number Triangle 1 (Middle Column - Top)
```

```
print("Pattern 4: Single Number Triangle")
```

```
for i in range(1, 6):
```

```
    for j in range(i):
```

```
        print(i, end="")
```

```
    print()
```

```
print("\n" + "="*30 + "\n")
```

Pattern 5: Letter Triangle

A

B B

C C C

D D D D

E E E E E

Pattern 5: Letter Triangle (Middle Column - Bottom)

```
print("Pattern 5: Letter Triangle")
```

```
for i in range(5):
```

```
    letter = chr(65 + i) # A=65, B=66, etc.
```

```
    for j in range(i + 1):
```

```
        print(letter, end=" ")
```

```
    print()
```

```
print("\n" + "="*30 + "\n")
```

Pattern 6: Sequential Number Triangle

0

01

012

0123

01234

012345

Pattern 6: Sequential Numbers (Left Column - Bottom)

```
print("Pattern 6: Sequential Number Triangle")
```

```
for i in range(6):
```

```
    for j in range(i + 1):
```

```
        print(j, end="")
```

```
    print()
```

```
print("\n" + "="*30 + "\n")
```

Pattern 7: Reverse Number Triangle

54321

4321

321

21

1

Pattern 7: Reverse Number Triangle (Right Column - Top)

```
print("Pattern 7: Reverse Number Triangle")
```

```
for i in range(5, 0, -1):
```

```
    for j in range(i, 0, -1):
```

```
        print(j, end="")
```

```
    print()
```

```
print("\n" + "="*30 + "\n")
```

Pattern 8: Right-aligned Star Triangle

```
*  
* *  
* * *  
* * * *  
* * * * *
```

Pattern 8: Right-aligned Star Triangle (Right Column - Middle)

```
print("Pattern 8: Right-aligned Star Triangle")
```

```
n = 5
```

```
for i in range(1, n + 1):
```

```
    # Print spaces
```

```
    for j in range(n - i):
```

```
        print(" ", end="")
```

```
    # Print stars
```

```
    for k in range(i):
```

```
        print("*", end=" ")
```

```
    print()
```

```
print("\n" + "="*30 + "\n")
```

Pattern 9: Spaced Number Triangle

1

2 1

3 2 1

4 3 2 1

5 4 3 2 1

Pattern 9: Number Pattern with Spaces (Right Column - Bottom)

```
print("Pattern 9: Spaced Number Triangle")
```

```
for i in range(1, 6):
```

```
    # Print spaces for alignment
```

```
    for j in range(5 - i):
```

```
        print(" ", end="")
```

```
    # Print numbers
```

```
    for k in range(i, 0, -1):
```

```
        print(k, end=" ")
```

```
    print()
```

```
print("\n" + "="*30 + "\n")
```

