

Question 1

Correct

Mark 1.00 out of 1.00

Problem Description:

Develop a Python program that safely calculates the square root of a number provided by the user. Handle exceptions for negative inputs and non-numeric inputs.

Input Format:

User inputs a number.

Output Format:

Print the square root of the number or an error message if an exception occurs.

For example:

Input	Result
16	The square root of 16.0 is 4.00
-4	Error: Cannot calculate the square root of a negative number.
rec	Error: could not convert string to float

Answer: (penalty regime: 0 %)

```
1 import math
2
3
4
5 try:
6     a=float(input())
7     if(a>=0):
8         n=math.sqrt(a)
9         print("The square root of",a,"is %.2f"%n)
10    else:
11        print("Error: Cannot calculate the square root of a negative number.")
12
13 except ValueError:
14     print("Error: could not convert string to float")
15
16
```

	Input	Expected	Got	
✓	16	The square root of 16.0 is 4.00	The square root of 16.0 is 4.00	✓
✓	0	The square root of 0.0 is 0.00	The square root of 0.0 is 0.00	✓
✓	-4	Error: Cannot calculate the square root of a negative number.	Error: Cannot calculate the square root of a negative number.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

Write a Python program that asks the user for their age and prints a message based on the age. Ensure that the program handles cases where the input is not a valid integer.

Input Format: A single line input representing the user's age.

Output Format: Print a message based on the age or an error if the input is invalid.

For example:

Input	Result
twenty	Error: Please enter a valid age.
25	You are 25 years old.
-1	Error: Please enter a valid age.

Answer: (penalty regime: 0 %)

```
1 try:
2     age = int(input())
3     if (age<0):
4         print("Error: Please enter a valid age.")
5     else:
6         print(f"You are {age} years old.")
7 except ValueError:
8     print("Error: Please enter a valid age.")
9 except Exception as e:
10    print("Error: Please enter a valid age.")
```

	Input	Expected	Got	
✓	twenty	Error: Please enter a valid age.	Error: Please enter a valid age.	✓
✓	25	You are 25 years old.	You are 25 years old.	✓
✓	-1	Error: Please enter a valid age.	Error: Please enter a valid age.	✓
✓	150	You are 150 years old.	You are 150 years old.	✓
✓		Error: Please enter a valid age.	Error: Please enter a valid age.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Problem Description:

Write a Python program that asks the user for their age and prints a message based on the age. Ensure that the program handles cases where the input is not a valid integer.

Input Format:

A single line input representing the user's age.

Output Format:

Print a message based on the age or an error if the input is invalid.

For example:

Input	Result
25	You are 25 years old.
rec	Error: Please enter a valid age.
-5	Error: Please enter a valid age.

Answer: (penalty regime: 0 %)

```

1 try:
2     a=int(input())
3     if a<0:
4         print("Error: Please enter a valid age.")
5     else:
6         print(f"You are {a} years old.")
7 except ValueError:
8     print("Error: Please enter a valid age.")
9 except Exception as b:
10    print("Error: Please enter a valid age.")

```

	Input	Expected	Got	
✓	25	You are 25 years old.	You are 25 years old.	✓
✓	rec	Error: Please enter a valid age.	Error: Please enter a valid age.	✓
✓	!@#	Error: Please enter a valid age.	Error: Please enter a valid age.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

Problem Description:

Write a Python script that asks the user to enter a number within a specified range (e.g., 1 to 100). Handle exceptions for invalid inputs and out-of-range numbers.

Input Format:

User inputs a number.

Output Format:

Confirm the input or print an error message if it's invalid or out of range.

For example:

Input	Result
1	Valid input.
101	Error: Number out of allowed range
rec	Error: invalid literal for int()

Answer: (penalty regime: 0 %)

```

1 try:
2     a=int(input())
3     if (a<=100 and a>=1):
4         print("Valid input.")
5     else:
6         print("Error: Number out of allowed range")
7 except ValueError:
8     print("Error: invalid literal for int()")

```

	Input	Expected	Got	
✓	1	Valid input.	Valid input.	✓
✓	100	Valid input.	Valid input.	✓
✓	101	Error: Number out of allowed range	Error: Number out of allowed range	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

Develop a Python program that safely performs division between two numbers provided by the user. Handle exceptions like division by zero and non-numeric inputs.

Input Format: Two lines of input, each containing a number.

Output Format: Print the result of the division or an error message if an exception occurs.

For example:

Input	Result
10 2	5.0
10 0	Error: Cannot divide or modulo by zero.
ten 5	Error: Non-numeric input provided.

Answer: (penalty regime: 0 %)

```

1 | try:
2 |     n,d=float(input()),float(input())
3 |     print(n/d)
4 | except ZeroDivisionError:
5 |     print("Error: Cannot divide or modulo by zero.")
6 | except ValueError:
7 |     print("Error: Non-numeric input provided.")

```

	Input	Expected	Got	
✓	10 2	5.0	5.0	✓
✓	10 0	Error: Cannot divide or modulo by zero.	Error: Cannot divide or modulo by zero.	✓
✓	ten 5	Error: Non-numeric input provided.	Error: Non-numeric input provided.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◀ [Week11_MCQ](#)

Jump to...