

8.13. Exercises

1.

Question

Answer

Write code that asks the user to enter a numeric score (0-100). In response, it should print out the score and corresponding letter grade, according to the table below.

Score	Grade
≥ 90	A
[80-90)	B
[70-80)	C
[60-70)	D
< 60	F

The square and round brackets denote closed and open intervals. A closed interval includes the number, and open interval excludes it. So 79.99999 gets grade C, but 80 gets grade B.

Save & Run

6/11/2020, 11:09:47 PM - 2 of 2

Show Feedback

Hide Code

Show in CodeLens

```
1 sc = input("Enter a score from 0 to 100 (decimal points)
2 fl_sc = float(sc)
3
4 if fl_sc < 60:
5     gr = "F"
6 elif fl_sc < 70:
7     gr = "D"
8 elif fl_sc < 80:
9     gr = "C"
10 elif fl_sc < 90:
11     gr = "B"
12 else:
13     gr = "A"
14
15 print("Score", fl_sc, "gets a grade of", gr)
```

Score 77.0 gets a grade of C

Activity: 1 -- ActiveCode (ac7_14_1)

2.

Question

A year is a **leap year** if it is divisible by 4. If the year can be evenly divided by 100, it is NOT a leap year, unless the year is **also** evenly divisible by 400. Then it is a leap year. Write code that asks the user to input a year and output True if it's a leap year, or False otherwise. Use if statements.

Year	Leap?
1944	True
2011	False
1986	False
1800	False
1900	False
2000	True
2056	True

Above are some examples of what the output should be for various inputs.

8.12. Glossary">

Glossary">

8.14. Chapter Assessment">

Save & Run6/12/2020, 11:34:51 PM - 12 of 12

Show FeedbackHide CodeHide CodeLens

```
1 year = input("Enter an year:")
2 print(year)
3 if (int(year)%4 == 0 or int(year)%100 == 0):
4     print(True)
5 elif (int(year)%100 == 0):
6     print(False)
7 else:
8     print(False)
9
10
```

1944
True

Python 3.6

```
1 year = input("Enter an year:")
2 print(year)
3 if (int(year)%4 == 0 or int(year)%100 == 0):
4     print(True)
5 elif (int(year)%100 == 0):
6     print(False)
7 else:
8     print(False)
```

Print output (drag lower right corner to resize)

Enter an year:2011
2011
False

Frames

Objects

Global frame
year "2011"

line that just executed

next line to execute

< Prev

Next >

Done running (5 steps)

[Python Tutor by Philip Guo](#)
[Customize visualization \(NEW!\)](#)

Activity: 2 -- ActiveCode (ac7_14_2)

3.

Question

Answer

What do these expressions evaluate to?

```
1. 3 == 3
2. 3 != 3
3. 3 >= 4
4. not (3 < 4)
```

Save & Run6/12/2020, 12:58:35 AM - 2 of 2

Show Feedback

Hide CodeShow in CodeLens

```
1 print(3 == 3)
2 print(3 != 3)
3 print(3 >= 4)
4 print(not(3 < 4))
5
```

True

```
True
False
False
False
```

Activity: 3 -- ActiveCode (ac7_14_3)

4.

Question

Give the **logical opposites** of these conditions, meaning an expression that would produce False whenever this expression produces True, and vice versa. You are not allowed to use the `not` operator.

1. `a > b`
2. `a >= b`
3. `a >= 18 and day == 3`
4. `a >= 18 or day != 3`

Save & Run

6/12/2020, 1:04:01 AM - 11 of 11

Show Feedback

Hide Code

Show in CodeLens

```
1 a = 18
2 b = 12
3 day = 3
4 print(a <= b)
5 print(a < b)
6 print(a<18 and day !=3)
7 print(a<18 or day !=3)
8 print(" ")
9 print(a > b)
10 print(a >= b)
11 print(a>=18 and day ==3)
12 print(a>=18 or day !=3)
```

```
False
False
False
False
```

```
True
True
True
True
```

Activity: 4 -- ActiveCode (ac7_14_4)

5.

Question

Provided are the lengths of two sides of a right-angled triangle. Assign the length of the hypotenuse to the variable `hypo_len`. (Hint: `x ** 0.5` will return the square root, or use `sqrt` from the `math` module)

Save & Run

6/12/2020, 1:07:44 AM - 4 of 4

Show Feedback

Hide Code

```
1 import math
2 side1 = 3
3 side2 = 4
4 x = (side1)*(side1)
5 y = (side2)*(side2)
6 hypo_len = math.sqrt(x+y)
7 print(hypo_len)
8
```

5.0

Activity: 5 -- ActiveCode (ac7_14_5)

Result	Actual Value	Expected Value	Notes
Pass	5.0	5	Testing that hypo_len has been set correctly

You passed: 100.0% of the tests

6.

Question

Provided is a list of numbers. For each of the numbers in the list, determine whether they are even. If the number is even, add `True` to a new list called `is_even`. If the number is odd, then add `False`.

Save & Run

6/12/2020, 1:10:37 AM - 4 of 4

Show Feedback

Hide Code

```
1 num_lst = [3, 20, -1, 9, 10]
2 is_even = []
3 for i in num_lst:
4     if (i%2 == 0):
5         is_even.append(True)
6     else:
7         is_even.append(False)
8 print(is_even)
9
```

[False, True, False, False, True]

Activity: 6 -- ActiveCode (ac7_14_6)

Result	Actual Value	Expected Value	Notes
Pass	[False...True]	[False...True]	Testing that is_even is set correctly.

Expand Differences

You passed: 100.0% of the tests

7.

Question

Provided is a list of numbers. For each of the numbers in the list, determine whether they are odd. If the number is odd, add `True` to a new list called `is_odd`. If the number is even, then add `False`.

Save & Run

6/12/2020, 1:11:22 AM - 2 of 2

Show Feedback

Hide Code

```
1 num_lst = [3, 20, -1, 9, 10]
2 is_odd = []
3 for i in num_lst:
4     if (i%2 != 0):
5         is_odd.append(True)
6     else:
```

```

7         is_odd.append(False)
8     print(is_odd)
9
10
11

```

```
[True, False, True, True, False]
```

Activity: 7 -- ActiveCode (ac7_14_7)

Result	Actual Value	Expected Value	Notes
Pass	[True...alse]	[True...alse]	Testing that is_odd is set correctly.

Expand Differences

You passed: 100.0% of the tests

8.

Question

Given the lengths of three sides of a triangle, determine whether the triangle is right angled. If it is, the assign `True` to the variable `is_rightangled`. If it's not, then assign `False` to the variable `is_rightangled`.

Hint: floating point arithmetic is not always exactly accurate, so it is not safe to test floating point numbers for equality. If a good programmer wants to know whether `x` is equal or close enough to `y`, they would probably code it up as

```

if abs(x - y) < 0.001:    # if x is approximately equal to y
    ...

```

Save & Run

6/12/2020, 11:31:14 PM - 8 of 8

Show Feedback

Hide Code

Show in CodeLens

```

10 if (abs(a-math.sqrt(y+z)) < 0.001):
11     is_rightangled = True
12 elif ((abs(b-math.sqrt(x+z)) < 0.001)):
13     is_rightangled = True
14 elif (c == math.sqrt(x+y)):
15     is_rightangled = True
16 else:
17     is_rightangled = False
18 print(is_rightangled)
19
20
21
22
23

```

```

25
False

```

Activity: 8 -- ActiveCode (ac7_14_8)

Result	Actual Value	Expected Value	Notes
Pass	False	False	Testing whether is_rightangled is set correctly

You passed: 100.0% of the tests

9.

Question

Answer

Discussion

Implement the calculator for the date of Easter.

The following algorithm computes the date for Easter Sunday for any year between 1900

to 2099.

Ask the user to enter a year. Compute the following:

1. $a = \text{year} \% 19$
2. $b = \text{year} \% 4$
3. $c = \text{year} \% 7$
4. $d = (19 * a + 24) \% 30$
5. $e = (2 * b + 4 * c + 6 * d + 5) \% 7$
6. $\text{dateofeaster} = 22 + d + e$

Special note: The algorithm can give a date in April. You will know that the date is in April if the calculation gives you an answer greater than 31. (You'll need to adjust) Also, if the year is one of four special years (1954, 1981, 2049, or 2076) then subtract 7 from the date.

Your program should print an error message if the user provides a date that is out of range.

Save & Run

6/12/2020, 1:29:09 PM - 6 of 6

Show Feedback

Hide Code

Show in CodeLens

```
1 year = input("Enter an year between 1900 to 2009:")
2 year = int(year)
3 spl_years = [1954,1981,2049,2076]
4 if (year in spl_years):
5     year -= 7
6 print(year)
7 a = year % 19
8
9 b = year % 4
10
11 c = year % 7
12
13 d = (19 * a + 24) % 30
14
15 e = (2 * b + 4 * c + 6 * d + 5) % 7
```

2009
43

Activity: 9 -- ActiveCode (ac7_14_9)

10.

Question

Get the user to enter some text and print out True if it's a palindrome, False otherwise. (Hint: Start by reversing the input string, and then use the == operator to compare two values to see if they are the same)

Save & Run

6/12/2020, 12:48:21 PM - 5 of 5

Show Feedback

Hide Code

Show in CodeLens

```
1 x = input("Enter something:")
2 print(x[::-1])
3 if (x == x[::-1]):
4     print("True")
5 else:
6     print("False")
7
8
```

retal
False

Activity: 10 -- ActiveCode (ac7_14_10)

11. condition-14-11: Write a program that will print out a greeting to each student in the list. This list should also keep track of how many students have been greeted and note that each time a new student has been greeted. When only one student has entered, the program should say "The first student has entered!". Afterwards, the program should say "There are {number here} students in the classroom!".

Drag from here

Drop blocks here

```
students = ["Jay", "Stacy", "Iman", "Trisha", "Ahmed", "Daniel", "Shadae", "Tosin", "Charlotte"]
num_students = 0
for student in students:
    print("Welcome to class, " + student)
    num_students += 1
    if num_students == 1:
        print("The first student has entered!")
    elif num_students > 1:
        print("There are " + str(num_students) + " students in the classroom!")
```

Check

Reset

Perfect! It took you only one try to solve this. Great job!

Activity: 11 -- Parsons (pp7_14_11)

12. condition-14-12: Piece together a program so that it can successfully print out one print statement, given the value of x.

Drag from here

Drop blocks here

```
x = 16
if x > 10:
    if x > 20:
        print("This is a large number!")
    else:
        print("This is a pretty big number.")
```

Check

Reset

Perfect! It took you only one try to solve this. Great job!

Activity: 12 -- Parsons (pp7_14_12)

8.12. Glossary">

Glossary">

8.14. Chapter Assessment">

>