

Computer Science 1 (../index.html) CPSC 1301K (../index.html)

Selection Practice Assignments

[Previous \(../functions/functionsPracticeAssignments.html\)](#)

Overview

Submission Instructions

For each of the following practice assignments, save your solution in a .py file, with the name being selections and the number of the assignment. For example, for "Selection01: isOdd", save your solution in a file named selection01.py (notice the lowercase "s"). Then, submit that file the respective assignment on codePost.io (<https://codePost.io>).

To register for a free account, go to <https://codepost.io/signup/join?code=I5039NQNWJ> (<https://codepost.io/signup/join?code=I5039NQNWJ>). Register with your CSU email address. Sometimes it takes more than one try, so please work on this well before the deadline. Additionally, some students have had better luck with using the "Forgot password" link.

Practice Assignments

Selection01: isOdd

Complete the isOdd() function to take an int value and returns True if the parameter's value is odd and False otherwise.

Examples:

isOdd(5) returns True

isOdd(10) returns False

Provided code:

```
def isOdd():  
  
  
print( isOdd( 5 ) )  
print( isOdd( 10 ) )
```

Note, after completing the function, the provided code should only display:

```
True  
False
```

Selection02: Division and Remainders

Complete the `canDivideNoRem()` function so that it takes a dividend and divisor and returns `True` if the division operation can be done without a remainder and `False` if not.

Examples:

`canDivideNoRem(44, 4)` returns `True`

`canDivideNoRem(20, 3)` returns `False`

Provided code:

```
def canDivideNoRem():
```

Selection03: Biggest Number

Complete the `biggest()` function so that it takes 4 numbers and returns the biggest one. Assume all values are unique (no duplicate numbers).

Examples:

`biggest(35, 32, 1, 9)` returns 35

`biggest(4, 9, 45, 3)` returns 45

Provided code:

```
def biggest():
```

Selection04: Smallest Number

Complete the `smallest()` function so that it takes 4 numbers and returns the smallest one. Assume all values are unique (no duplicate numbers).

Examples:

`smallest(35, 32, 1, 9)` returns 1

`smallest(4, 9, 45, 3)` returns 3

Provided code:

```
def smallest():
```

Selection05: Middle Number

Complete the `middle()` function so that it takes 3 numbers and returns the middle value.
Assume all values are unique (no duplicate numbers)

Examples:

`middle(35, 1, 32)` returns 32

`middle(14, 9, 45)` returns 14

Provided code:

```
def middle():
```

Selection06: Letter Grade

Complete the `letterGrade()` function so that it takes a number score and returns a letter grade. Use the following scale:

90 or above: 'A'

80-89: 'B'

70-79: 'C'

60-69: 'D'

<60: 'F'

Examples:

`letterGrade(75)` returns 'C'

`letterGrade(98)` returns 'A'

`letterGrade(55)` returns 'F'

Provided code:

```
def letterGrade():
```

Selection07: Zip Zap Zop

Complete the `zipZapZop()` function so that it requests an integer from the user. The function should display a response following this pattern:

- Divisible by 3: zip
- Divisible by 5: zap
- Divisible by 7: zop

Otherwise, just display the number.

Note 1: numbers that are divisible by more than one (3, 5, or 7) should contain all applicable terms.

Notes 2: Note that you are displaying directly from this function, not returning a value to the calling function.

Example 1:

```
Enter a number: 5
zap
```

Example 2:

```
Enter a number: 15
zipzap
```

Example 3:

```
Enter a number: 2
2
```

Hint: The number only prints if it is not divisible by ALL of 3, 5, or 7.

Provided code:

```
def zipZapZop():
```

Selection08: Speeding Ticket

Complete the `speeding()` function so that it takes 1) a speed and 2) a speed limit. Have the function return the total fine according to the following:

If someone is caught speeding, it is a \$50 fine plus \$4 for each MPH over the speed limit.

If the speed is 90 MPH or more, add another \$150 fine.

If there is 25 MPH or more difference between the speed and speed limit add another \$300 fine.

If someone was not speeding, the function should return: No speed violation

Examples:

speeding(75, 60) returns 110

speeding(90, 75) returns 260

speeding(50, 25) returns 450

speeding(35, 40) returns 'No speed violation'

Provided code:

```
def speeding():
```