SOLUTION CP1200 – Practical 3

Aim

1. Practise writing Python programs that use decision structures (if, if-else, if-elif-else)
2. Practise using functions and other code constructs

Decision Structures and Simple Functions

Problems For You To Fill In The Blanks

1. Print odd or even

```
"""
prints "odd" or "even" based on number
remember, number % 2 gives 0 when number is divisible by 2, and 1 otherwise
"""
def printOddOrEven(number):
    if number % 2 == 1:
        print("odd")
    else:
        print("even")

print("6 is", end=' ')
printOddOrEven(6)
print("7 is", end=' ')
printOddOrEven(7)
print("8 is", end=' ')
printOddOrEven(8)
```

The output of the above code should be:

```
6 is even
7 is odd
8 is even
```

2. Display tax

The Python Party wins government at the next election and introduce a simpler tax system that works like this:

- If you earn under $100, you pay no tax
- If you earn between $100 and $1000, you pay 5% tax on the total amount
- If you earn over $1000, you pay 10% tax on the total amount

Remember the tutorial... there are three cases (and three conditions, with three actions). Write the table or pseudocode for this, then code up the solution, some of which is started for you below. Note the reuse of a function we wrote last week, and the good use of constants. Make sure to only put your calculation for takeHomePay in one spot - it doesn't need to go in each path of the if-elif-else statement.

```
TAX_RATE_LOW = 0.05 # 5%
TAX_RATE_HIGH = 0.1 # 10%
TAX_THRESHOLD_LOW = 100
TAX_THRESHOLD_HIGH = 1000

print("Python Party Tax Program - Where Tax is a Party")
income = float(input("Income: $"))
taxAmount = 0 # we need to declare this variable here so we can use it later
if income < TAX_THRESHOLD_LOW:
      taxAmount = 0
elif income < TAX_THRESHOLD_HIGH:
      taxAmount = TAX_RATE_LOW * income
else:
      taxAmount = TAX_RATE_HIGH * income
takeHomePay = income - taxAmount
print("Total tax is: ", end="")
printCurrency(taxAmount)
print("Take home pay is: ", end="")
printCurrency(takeHomePay)
```

Here is some sample output for two runs of the program:

```
Python Party Tax Program - Where Tax is a Party
Income: $10000
Total tax is: $1000.00
Take home pay is: $9000.00
```

```
Python Party Tax Program - Where Tax is a Party
Income: $595
Total tax is: $29.75
Take home pay is: $565.25
```

Programming Problems

ShipShape…

Here's some example output (`green` represents user input):

```
How many of shape A: -1
Invalid quantity, using 0
How many of shape B: 10
Total cost is: $108.00
```

SOLUTION – See Python File

Debugging

1. Someone (it's not polite to say who), was trying to write a program to tell the user if their score is invalid, passable, excellent or bad, but their code is in the "bad" category and doesn't work. Rewrite the following programming attempt using the most efficient if-elif-else 'ladder' you can find:

```
if score < 0:
    print("Invalid score")
else:
    if score > 100:
        print("Invalid score")
    if score > 50:
        print("Passable")
    if score > 90:
    print("Excellent")
if score < 50:
    print("Bad")
```

SOLUTION – Notice that we assume that 90 = excellent and 50 = passable, so we use >=

```
if score < 0 or score > 100:
    print("Invalid score")
elif score >= 90:
    print("Excellent")
elif score >= 50:
    print("Passable")
else:
    print("Bad")
```

Form Letter - Version 2

SOLUTION – See Python file

Extension

1. We wrote printOddOrEven earlier, which basically determines whether a number is divisible by 2. Write a function that takes two parameters, and prints "Divisible" is the first is divisible by the second, and "Not divisible" otherwise. Give the function an appropriate name.

SOLUTION

```
def printDivisible(x, y):
    if x % y == 0:
        print("Divisible")
    else:
        print("Not divisible")
```

2. Add a "discount applied" message to the ShipShape program. The program should run and determine the total shipping cost, stored in the one variable, and if a discount was applied, it should also print (in addition to the usual outputs) "10% discount applied". Don't use a literal for "10%". It should be based on your constant, so that if this changes, the output will print (e.g. "17.5% discount applied").

SOLUTION – see Python file