Sec 4 Computing Prelim 2019 P2 Ans Scheme

1	E3=TODAY()	[1m]
2	E12=ROUND(B13/(B12^2),1) OR ROUND(B13/(B12*B12),1) OR ROUND(B13/POWER(B12,2),1) -1m if not shown in 1dp or brackets missing	[2m]
3	E13 =VLOOKUP(E12,D25:E27,2,TRUE)	[1m]
4	E15=IF(AND(B14<120,B15<80),E31,IF(AND(B14>140,B15>90),E33,E32) -1m if not referenced using cell address	[2m]
5	E20 =SUM(B20, B21) OR SUM(B20:B21)	[1m]
6	 E21= ROUND(B21/E20,2) OR B21/E20 with formatting to zero decimal place -1m if not shown in percentage/whole numbers -1m if not referenced using cell address 	[2m]
7	Conditional Formatting Rules Manager ? X	[1m]
7	Conditional Formatting Rules Manager ? × Show formatting rules for: Current Selection	[1m]
7	Conditional Formatting Rules Manager ? × Show formatting rules for: Current Selection New Rule Edit Rule	[1m]
7	Conditional Formatting Rules Manager ? × Show formatting rules for: Current Selection New Rule Edit Rule Edit Rule Delete Rule Rule (applied in order shown) Format Applies to Stop If True Cell Value < 0.5 AaBbCcYyZz	[1m]
7	Conditional Formatting Rules Manager ? × Show formatting rules for: Current Selection Image: New Rule Image: Edit Rule Pelete Rule Image: Pelete Rule Rule (applied in order shown) Format AaBbCcYyZz Stop If True Image: Pelete Rule Rule (applied in order shown) Format Applies to Stop If True Image: Pelete Rule Image: Pelete Rule	[1m]
7	Conditional Formatting Rules Manager ? × Show formatting rules for: Current Selection Image: New Rule Image: Delete Rule Rule (applied in order shown) Format Cell Value < 0.5	[1m]

```
number = int(input("Enter a positive integer: "))
   while int(number)<=0:</pre>
8b
                                                                        [1m]
        number=int(input("Re-enter a positive integer: "))
                                                                        [1m]
   count=0
8a
   print("Factors of this integer are: ")
   for factor in range(2, number):
                                                                        [1m]
        if number % factor == 0:
            count+=1
            print(factor)
                                                                        [1m]
   print("There are a total of {} factors.".format(count))
   number = int(input("Enter a positive integer: "))
   print("Factors of this integer are: ")
9b
                                                                        [3m]
   for factor in range(2, int(number**0.5)+1):
        if number % factor == 0:
9a
                                                                        [3m]
            print(factor, "x", number//factor)
   9a:
   1m - find corresponding factor using floor division
   1m – proper output, line by line
   1m – output both
   9b:
   1m – identify repeated factors or use sqrt,
   1m – remove repeated,
   1m – exclude 1 and itself
```

```
10 print(" ~~Drinks Menu~~ ")
   drinks=["Hot Coffee", "Hot Tea", "Canned Drink", "Bottled
   Drink"]
   prices=[4,3,2.5,3.5]
                                                                    [1m]
   for option in range (len (drinks)):
                                                                    [1m]
       print(str(option+1)+"-
   "+drinks[option]+"\t$"+str(prices[option]))
                                                                    [1m]
   choice=int(input("\nEnter your choice: "))
                                                                    [1m]
   owe=prices[choice-1]
   print("\n"+drinks[choice-1]+" costs $"+str(owe)+".")
   paid=0
                                                                    [1m]
   valid=["0.1","0.2","0.5","1","2","5","10"]
                                                                    [1m]
   while paid<owe:
       cash=input("Please insert cash(\"x\" to cancel): ")
       if cash in valid:
                                                                    [1m]
           paid+=float(cash)
                                                                    [1m]
       Helif 'cash=="x":
                                                                    [1m]
           print("Your order is cancelled.")
           owe=0
      Jelse:
                                                                    [1m]
           print("Invalid currency!")
       print("\nBalance payment is $"+str(owe-paid)+".")
   if paid>owe:
      print("\n$"+str(paid-owe)+" has been returned to you.")
   if cash!="x":
                                  D
       print("Your "+drinks[choice-1]+" has been served.
   Enjoy!")
```

11	Ask user for an input date (e.g. DD/MM/YYYY) and store it appropriately.	[1m]
	Validation if input has D, M and Y being digits and length is 10.	[1m]
	Use while-loop to ask user to re-enter input if valid	[1m]
	Separate DD, MM and YYYY using indexing or slicing and store in appropriate variables	[1m]
	Use mod (%) in conditions to determine leap year	[1m]
	Use if-else logically to determine leap year	[1m]
	Use list to store data on how many days in a month, and $1 \le month \le 30$	[1m]
	Use if-else logically to determine if day corresponds to month, e.g. $1 \le day \le 30$	[1m]
	Use leap year to decide if 29 Feb is valid	
	Output appropriately to inform user if it is leap year and/or if date is valid	
	(Any 8 points to score maximum 8m)	
	-1m for poor programming habits, e.g. improper variable name	
	-1m for any syntax error that disrupts the proper running of the program	
	-1m for any logical error that contradicts the program requirements in the question	
12	Input prompts are appropriate, e.g. ask for DD/MM/YYYY format	[1m]
	Date validity output correctly and appropriately	[1m]
13	Initialize counter for counting leap years	[1m]
	For-loop or while-loop with appropriate range, 1582 up to previous year	[1m]
	Increasing counter logically each time the loop is running	[1m]
	Output leap year results with an appropriate statement	[1m]
14	Initialize days to store total number of days since 1 Jan 1583, add days in current month	[1m]
	Calculate total days from 1 Jan 1583 to 1 Jan of current year assuming 365 days per year	[1m]
	Add extra days during leap years into the total days	[1m]
	For-loop to add total number of days in months of the current year up to current month	[1m]
	Using if-else appropriately to determine if date in current year has extra day	[1m]
	Use list to store names of the days of the week, in the correct order	[1m]
	Use mod (%) to determine which day of the week and output correct day of the week	
	(Any 6 points to score maximum 6m)	