Anglo-Chinese School (Barker Road)

SECONDARY FOUR EXPRESS

COMPUTING

PRELIMINARY EXAMINATION

Marking Scheme
Paper 2

Task 1 MYCUSTOMER

	A	B	C	D	E
1	Cost for Customers				
2	ID	Type	Cost	After Discount	Final Cost
3	64S97	Staff	$44.76	$38.05	$38.00
4	96M33	Member	$13.83	$12.45	$12.00
5	59N85	Non-Member	$4.26	$4.26	$4.00
6	35N53	Non-Member	$6.65	$6.65	$6.00
7	58N85	Non-Member	$73.79	$73.79	$73.00
8	73M15	Member	$25.24	$22.72	$22.00
9	11M87	Member	$67.61	$60.85	$60.00
10	96M81	Member	$36.44	$32.80	$32.00
11	51M85	Member	$93.11	$83.80	$83.00
12	76S63	Staff	$5.86	$4.98	$4.00
13	30S75	Staff	$66.71	$56.70	$56.00
14	89M11	Member	$49.66	$44.69	$44.00
15	80N70	Non-Member	$51.96	$51.96	$51.00
16	64M99	Member	$69.14	$62.23	$62.00
17	99S43	Staff	$11.32	$9.62	$9.00
18	71S31	Staff	$21.35	$18.15	$18.00
19	36M16	Member	$40.53	$36.48	$36.00
20	88N27	Non-Member	$21.60	$21.60	$21.00
21					
22	Type of Customers				
23	Type	Member	Non-Member	Staff	
24	Discount Rate	10.00%	0.00%	15.00%	
25	Total Final Cost	$351.00	$155.00	$125.00	

	A	B	C	D	E
1	Cost for Customers				
2	ID	Type	Cost	After Discount	Final Cost
3	64S97	=IF(MID(A3,3,1)="S", "Staff", IF(MID(A3,3,1)="M", "Member", "Non-Member"))	44.76	=(1-HLOOKUP(B3,$A$23:$D$24,2,TRUE))*C3	=FLOOR(D3,1)
4	96M33	=IF(MID(A4,3,1)="S", "Staff", IF(MID(A4,3,1)="M", "Member", "Non-Member"))	13.83	=(1-HLOOKUP(B4,$A$23:$D$24,2,TRUE))*C4	=FLOOR(D4,1)
5	59N85	=IF(MID(A5,3,1)="S", "Staff", IF(MID(A5,3,1)="M", "Member", "Non-Member"))	4.26	=(1-HLOOKUP(B5,$A$23:$D$24,2,TRUE))*C5	=FLOOR(D5,1)
6	35N53	=IF(MID(A6,3,1)="S", "Staff", IF(MID(A6,3,1)="M", "Member", "Non-Member"))	6.65	=(1-HLOOKUP(B6,$A$23:$D$24,2,TRUE))*C6	=FLOOR(D6,1)
7	58N85	=IF(MID(A7,3,1)="S", "Staff", IF(MID(A7,3,1)="M", "Member", "Non-Member"))	73.79	=(1-HLOOKUP(B7,$A$23:$D$24,2,TRUE))*C7	=FLOOR(D7,1)
8	73M15	=IF(MID(A8,3,1)="S", "Staff", IF(MID(A8,3,1)="M", "Member", "Non-Member"))	25.24	=(1-HLOOKUP(B8,$A$23:$D$24,2,TRUE))*C8	=FLOOR(D8,1)
9	11M87	=IF(MID(A9,3,1)="S", "Staff", IF(MID(A9,3,1)="M", "Member", "Non-Member"))	67.61	=(1-HLOOKUP(B9,$A$23:$D$24,2,TRUE))*C9	=FLOOR(D9,1)
10	96M81	=IF(MID(A10,3,1)="S", "Staff", IF(MID(A10,3,1)="M", "Member", "Non-Member"))	36.44	=(1-HLOOKUP(B10,$A$23:$D$24,2,TRUE))*C10	=FLOOR(D10,1)
11	51M85	=IF(MID(A11,3,1)="S", "Staff", IF(MID(A11,3,1)="M", "Member", "Non-Member"))	93.11	=(1-HLOOKUP(B11,$A$23:$D$24,2,TRUE))*C11	=FLOOR(D11,1)
12	76S63	=IF(MID(A12,3,1)="S", "Staff", IF(MID(A12,3,1)="M", "Member", "Non-Member"))	5.86	=(1-HLOOKUP(B12,$A$23:$D$24,2,TRUE))*C12	=FLOOR(D12,1)
13	30S75	=IF(MID(A13,3,1)="S", "Staff", IF(MID(A13,3,1)="M", "Member", "Non-Member"))	66.71	=(1-HLOOKUP(B13,$A$23:$D$24,2,TRUE))*C13	=FLOOR(D13,1)
14	89M11	=IF(MID(A14,3,1)="S", "Staff", IF(MID(A14,3,1)="M", "Member", "Non-Member"))	49.66	=(1-HLOOKUP(B14,$A$23:$D$24,2,TRUE))*C14	=FLOOR(D14,1)
15	80N70	=IF(MID(A15,3,1)="S", "Staff", IF(MID(A15,3,1)="M", "Member", "Non-Member"))	51.96	=(1-HLOOKUP(B15,$A$23:$D$24,2,TRUE))*C15	=FLOOR(D15,1)
16	64M99	=IF(MID(A16,3,1)="S", "Staff", IF(MID(A16,3,1)="M", "Member", "Non-Member"))	69.14	=(1-HLOOKUP(B16,$A$23:$D$24,2,TRUE))*C16	=FLOOR(D16,1)
17	99S43	=IF(MID(A17,3,1)="S", "Staff", IF(MID(A17,3,1)="M", "Member", "Non-Member"))	11.32	=(1-HLOOKUP(B17,$A$23:$D$24,2,TRUE))*C17	=FLOOR(D17,1)
18	71S31	=IF(MID(A18,3,1)="S", "Staff", IF(MID(A18,3,1)="M", "Member", "Non-Member"))	21.35	=(1-HLOOKUP(B18,$A$23:$D$24,2,TRUE))*C18	=FLOOR(D18,1)
19	36M16	=IF(MID(A19,3,1)="S", "Staff", IF(MID(A19,3,1)="M", "Member", "Non-Member"))	40.53	=(1-HLOOKUP(B19,$A$23:$D$24,2,TRUE))*C19	=FLOOR(D19,1)
20	88N27	=IF(MID(A20,3,1)="S", "Staff", IF(MID(A20,3,1)="M", "Member", "Non-Member"))	21.6	=(1-HLOOKUP(B20,$A$23:$D$24,2,TRUE))*C20	=FLOOR(D20,1)
21					
22	Type of Customers				
23	Type	Member	Non-Member	Staff	
24	Discount Rate	0.1	0	0.15	
25	Total Final Cost	=SUMIF($B$3:$B$20,B23,$E$3:$E$20)	=SUMIF($B$3:$B$20,C23,$E$3:$E$20)	=SUMIF($B$3:$B$20,D23,$E$3:$E$20)	

Question	Answer	Marks
1	**One** mark for correct working conditional IF formula (in cell B3), **One** mark for the rest =IF(MID(A20,3,1)="S", "Staff", IF(MID(A20,3,1)="M", "Member", "Non-Member"))	[1] [1]
2	**One** mark for correct working HLOOKUP formula (in cell D3), **One** mark for the rest =(1-HLOOKUP(B3,$A$23:$D$24,2,TRUE))*C3	[1] [1]
3	**One** mark for correct working ROUND formula (in cell F3), **One** mark for the rest =FLOOR.Math(D3,1)	[1] [1]
4	**One** mark for correct working SUMIF formula (in cell B23), **One** mark for the rest =SUMIF($B$3:$B$20,B23,$E$3:$E$20)	[1] [1]
5	**One** mark for yellow fill for cells with staff, **One** mark for whole row. =$B3="Staff"	[1] [1]

Task 2 MYFIB

```
n1 = 0
n2 = 1
fiblist = [] #8

while True: #7
    nterms = int(input("Enter the nterm(s): ")) #6
    if nterms > 0: #7
        break
    else:
        print("nterms must be a positive integer.") #7

for i in range(nterms):
    print(n1)
    fiblist += [n1] #8
    nth = n1 + n2
    n1 = n2
    n2 = nth

print(fiblist)#8
```

Task 2 INFIB

```
n1 = 0
n2 = 1
fiblist = []

nterms = 100

for i in range(nterms):
    fiblist += [n1]
    nth = n1 + n2
    n1 = n2
    n2 = nth

print(fiblist)

num = int(input("Enter the number: "))#9
if num in fiblist:#9
    print("Yes, it is in the 100th sequence")#9
else:
    print("No, it is not in the 100th sequence")
```

Question	Answer	Marks
6	**One** mark for input of nterms with appropriate message	[1]
7	**One** mark for using while loop **One** mark for correct condition **One** mark for appropriate error message	[1] [1] [1]
8	**One** mark for initialisation list **One** mark for adding terms in the list **One** mark for printing the list	[1] [1] [1]

9	**One** mark for input of the positive integer **One** mark for checking if the number is in the 100th list **One** mark for output message is appropriate	[1] [1] [1]

Task 3

```
1   while True:
2       date = input("Enter the date (DD-MM-YYYY): ") #1
3       test = date
4       if len(test)== 10 and test[2]=="-" and test[5]=="-": #2
5           day = int(test[0:2])
6           month = int(test[3:5]) #3
7           year = int(test[6:])
8           check_year = year>1900 and year<=2022 #4
9           check_month = month>=1 and month<=12 #5
10          check_day_31 = day<=31 and (month in [1,3,5,7,8,10,12])
11          check_day_30 = day<=31 and (month in [4,6,9,11])
12          check_day_Feb = month == 2 and ((day<=29 and year%4==0) or day<=28)#6
13          if check_year: #7
14              if check_month:
15                  if check_day_31 or check_day_30 or check_day_Feb: #8
16                      break
17                  else:
18                      print("Error in day")
19              else:
20                  print("Error in month") #9
21          else:
22              print("Error in year") #9
23      else:
24          print("Error in format") #10
25  print("Date accepted")
```

Question	Answer	Marks
10	**One** mark for correct indent (line 2)	[1]
	One mark for using == to compare `len(test)` instead of `=` (line 4)	[1]
	One mark for correct string slicing `[3:5]` (line 6)	[1]
	One mark for correct current year `2022` instead of `2000` (line 8)	[1]
	One mark for using `and` instead of `or` (line 9)	[1]
	One mark for changing `month == 2` (line 12)	[1]
	One mark for adding _ for variable `check_year` (line 13)	[1]
	One mark for adding `:` at the end of `if` (line 15)	[1]
	One mark for swapping the "`month`" and "`year`" in the output (line 20 & 22)	[1]
	One mark for adding " for the output (line 24)	[1]

Task 4

```
def get_price(item):
    cake_list = ["A", "B", "C", "D","E","F","G","H"]
    price_list = [25, 22, 38, 35, 15, 40, 53, 20]
    position = cake_list.index(item)
    return price_list[position]

def get_input():
    while True:
        choice = input("Enter the choice of cake: ")
        if choice in ["A", "B", "C", "D","E","F","G","H"]:
            break
        else:
            print("Enter an uppercase letter between A to H only")
    return choice

def get_order():
    total = 0
    while True:
        cake = get_input()
        total += get_price(cake)

        cont = input("More purchase? Y or N: ")
        if cont == "N":

            GST = 0.07
            GST_amt = round(GST*total,2)
            GST_list = str(GST_amt).split(".")
            cents = GST_list[1]
            if int(cents[-1])>=5:
                final_GST = str(GST_list[0])+"."+str(GST_list[1])[0]+"5"
            else:
                final_GST = str(GST_list[0])+"."+str(GST_list[1])[0]+"0"

            final = total + float(final_GST)
            final_list = str(final).split(".")
            if len(final_list[1])== 1:
                final_list[1]+="0"
                final_display = final_list[0]+"."+final_list[1]
            else:
                final_display = str(final)
            print("")
            print("Subtotal \t" + "$ " + str(total))
            print("GST \t\t" + "$ " + final_GST)
            print("Total \t" + "$ " + final_display)
            print("")
            print("Thank you!")
            break

def get_order2():
    total = 0
    stock_list = [2,2,2,2,2,2,2,2]
    available = True
    cake_list = ["A", "B", "C", "D","E","F","G","H"]
    while True:
        cake = get_input()
        index = cake_list.index(cake)
        if stock_list[index]>=1:
            available = True
            stock_list[index]-=1
        else:
            available = False
```

```
if available == True:
    total += get_price(cake)
else:
    print("The cake is not available")
cont = input("Another purchase? Y or N: ")
if cont == "N":
    GST = 0.07
    GST_amt = round(GST*total,2)
    GST_list = str(GST_amt).split(".")
    cents = GST_list[1]
    if int(cents[-1])>=5:
        final_GST = str(GST_list[0])+"."+str(GST_list[1])[0]+"5"
    else:
        final_GST = str(GST_list[0])+"."+str(GST_list[1])[0]+"0"

    final = total + float(final_GST)
    final_list = str(final).split(".")
    if len(final_list[1])== 1:
        final_list[1]+="0"
        final_display = final_list[0]+"."+final_list[1]
    else:
        final_display = str(final)
    print("")
    print("Subtotal \t" + "$ " + str(total))
    print("GST \t\t" + "$ " + final_GST)
    print("Total \t" + "$ " + final_display)
    print("")
    print("Thank you!")
    break
```

Question	Answer	Marks
11	Allow user input the choice of cake	[1]
	Correct condition to make sure only A to H is input	[1]
	Return the choice of cake	[1]
12	Able to use of `get_input()`	[1]
	Able to use `get_price(option)`	[1]
	Able to allow more input(s) of the choice of cake	[1]
	Able to loop until no more additional cake is ordered	[1]
	Able to calculate the subtotal	[1]
	Able to calculate the GST amount	[1]
	Able to calculate the final total	[1]
	Able to display the subtotal	[1]
	Able to display the GST amount	[1]
	Able to display the GST amount to the nearest $0.05	[1]
	Able to display the final total	[1]
	Able to display the final total to the nearest $0.05	[1]
	Able to display the outputs aligned with $	[1]
	Able to display the output as the sample execution completely	[1]
13	Able to initialise the quantity for each type of cakes	[1]
	Able to update the quantity of the chosen type of cake	[1]
	Able to display warning message when the chosen type of cake is not available	[1]

End of Answer Key