Programming for Business Analytics – Mid Term Solutions

Nov 1, 2026

Question 1:

A while loop suits this problem, because the number of years is not known in advance.

(a)

Part	Meaning	In this problem
Initialization	Set the starting values before the loop	`balance = 50000`, `years = 0`
Condition	Tested before every pass; the loop runs while it is True	`balance < 120000`
Update	Changes a variable so the loop moves toward ending	`balance = balance * 1.13` and `years += 1`

A `for` loop is for a known number of repetitions, so it fits worse here.

```
balance = 50000
years = 0
while balance < 120000:
    balance = balance * 1.13
    years += 1
print(years, balance)
```

This prints 8 years with a balance of about Tk 132,922. After 7 years the balance is about Tk 117,631, still below 120,000.

(b)

Name	Valid?	Reason
`net_profit`	Valid	Letters and underscore only
`3rd_quarter`	Invalid	Cannot start with a digit
`unit cost`	Invalid	Cannot contain a space
`sales-tax`	Invalid	The hyphen is read as the minus operator; only letters, digits and _ are allowed
`_count`	Valid	A name may start with an underscore
`if`	Invalid	It is a reserved keyword

Question 2:

The program prints four lines and the customer's tier is Gold.

www.bba-archive.com

(a) Trace

- `subtotal = 250 * 4 = 1000`
- `total = 1000 - 1000 * 10 / 100 = 900.0` (a float, because `/` always gives a float)
- `points = int(900.0) // 100 = 900 // 100 = 9`
- `code = "BDT" + "9" = "BDT9"`
- `total % 300 = 900.0 % 300 = 0.0`
- `900.0 >= 900` and `9 > 8` are both True, so the first branch runs

Output:

```
900.0
9 0.0
BDT9
Gold
```

(b) With and without int()

	Value	Data type
With `int()`: `int(900.0) // 100`	9	int
Without `int()`: `900.0 // 100`	9.0	float

The printed tier would not change. `9.0 > 8` is still True and `total >= 900` is still True, so the program still prints Gold. The `code` string would change, though: `str(9.0)` gives `"BDT9.0"`, so `code` would be `"BDT9.0"` instead of `"BDT9"`.

Question 3: Five errors

The program has five errors, listed below with the reason and the corrected line for each.

#	Line	Reason	Corrected line
1	`qty = input("Quantity: ")`	`input()` returns a string, so `qty * price` raises a TypeError (str × float)	`qty = int(input("Quantity: "))`
2	`price = float(input("Unit price: ")`	The closing parenthesis is missing (SyntaxError)	`price = float(input("Unit price: "))`
3	`if member = "y":`	`=` is assignment; comparison needs `==`	`if member == "y":`
4	`else`	The colon is missing	`else:`
5	`print("Amount payable: " + payable)`	A string cannot be concatenated with a float	`print("Amount payable: " + str(payable))`

Question 4:

The program uses nested if–else statements; the two test cases give Tk 90 and Tk 120.

www.bba-archive.com

(a)

```
location = input("Location (inside/outside): ")
weight = float(input("Weight in kg: "))
premium = input("Premium customer (y/n): ")

if location == "inside":
    if weight <= 1:
        charge = 60
    else:
        charge = 60 + (weight - 1) * 20
else:
    if weight <= 1:
        charge = 120
    else:
        charge = 120 + (weight - 1) * 30

if premium == "y":
    charge = charge - charge * 10 / 100

print("Delivery charge:", charge)
```

Check against the samples: outside, 2.5 kg, premium gives 120 + 1.5 × 30 = 165, then 165 × 0.9 = 148.5. Inside, 0.8 kg, regular gives 60. Both match.

(b) Test cases

Case	Base charge	Discount	Printed output
(i) Inside Dhaka, 3 kg, premium	60 + (3 − 1) × 20 = 100	10% of 100 = 10	Delivery charge: 90.0
(ii) Outside Dhaka, 0.5 kg, regular	0.5 ≤ 1, so 120	None (not premium)	Delivery charge: 120

www.bba-archive.com