Assignment 0: Chai Corner

Fakhir Shaheen · Sections A and B

Weightage: ***1.5%***

Deadline: Thursday 17th September 2026, **11:59 PM**

Introduction

Chai Corner is a small tea shop on campus. Students go there between classes. The owner writes every bill by hand. By 11 in the morning, nobody can read the bills. You will write a program that prints a clean *receipt* (the paper bill you get after paying). The receipt shows the items, the total, the change, and the stamps on the customer's loyalty card.

What you will practise

- Reading what the user types with `input()`, and turning it into a number with `int()`.
- Making lines of text: joining pieces with `+`, repeating with `*`, cutting a piece out with `[ ]`, and turning a number into text with `str()`.
- Writing `True`/`False` expressions with `==`, `>=`, `and`, `or` (Lecture 4).
- Checking your own program with the test file we give you.

Collaboration

Students are NOT permitted to look at or copy each other's code or 'code structure' / logic. Seeking "Help" from seniors or friends is also prohibited. If caught, students will receive a 0 in A0.

AI tools (ChatGPT, Claude, Gemini, Copilot, or any other) are NOT allowed in this assignment. Not for writing code. Not for fixing code. Not for explaining errors. Not for the dev notes. Do not paste this handout or your file into any AI tool. If you use an AI tool in any way, it is plagiarism and you get 0 in A0.

Getting Started

1. Download `a0.py` and `test_a0_student.py`. **Save both files in the same folder.** Open `a0.py` in Thonny and read it. The top part is the *cover sheet* (like the front page of an exam paper). Below it, every part has a comment that tells you what to print. Part 1 is already started for you.
2. Fill in the cover sheet: your name, your roll number, and your name again after `Confirmed by:`. Just below the cover sheet are two lines, `CASHIER = ...` and `ROLL = ....` Write your name and your roll number there too, inside the quotes. At the end, write the *dev notes*: 3 to 5 sentences about the line that was hardest for you.

3. Open `test_a0_student.py` and press F5. It prints a report. Every check says PASS or FAIL. If a line is wrong, it shows the line we expected and the line you printed. This report is your to-do list. Run it again after every part.

Rule 1. Print every line exactly as shown. Look carefully at the spaces: one space after every colon `:`, and one space before and after `x` and `=`.

Rule 2. Money is in whole rupees. Use `//` (whole-number division), not `/`. So 5% of 390 is 19, not 19.5.

Rule 3. Use only what we did in Lectures 1 to 4: variables, `int()` and `str()`, `+ - * // %`, joining text with `+`, repeating text with `*`, cutting text with `[ ]` and `[ : ]`, `len()`, `print()`, `input()`, comparisons, and `and`/`or`/`not`. Do NOT use `if` (that is Lecture 5), loops, functions, lists, or anything with a dot like `.upper()`. The test file will tell you if you used something else. Also: every block of code (lines with no empty line between them) must start with a comment, a line beginning with `#`, that says what the block does.

Tip. Typing 10 inputs every time you run the program is boring. While you work, you can write `name1 = "Samosa"` instead of `name1 = input(...)`. Before you submit, change it back to `input(...)`. The test file needs the real `input()` lines.

The Receipt

Your program asks for 10 things, in this order: the name, price and quantity of item 1; then the same for item 2; then for item 3; then the cash the customer gives. Names have 1 to 20 letters. Prices are 1 to 9999. Quantities are 0 to 99. The cash is always enough to pay the total. Below is the full receipt when the cashier is `Sara Ahmed`, roll number `271050784`, and the customer buys 2 Masala Chai at 60 each, 3 Samosa at 40 each, 1 Chicken Roll at 150, and gives 500 rupees:

```
==============================
CHAI CORNER
FC College, Lahore
==============================
Cashier: Sara Ahmed
Till: 784
Receipt no: 1105
------------------------------
Masala Chai: 2 x 60 = 120
Samosa: 3 x 40 = 120
Chicken Roll: 1 x 150 = 150
------------------------------
Subtotal: 390
Discount 5%: 19
TOTAL: 371
Cash: 500
Change: 129
------------------------------
Stamps: [*][ ][ ][ ][ ]
Free chais earned: 1
Combo: True
Big order: False
Chai on order: True
Exact cash: False
==============================
Thank you, come again!
```

Every line of this receipt uses one of three things you already know:

1. **Repeat text with `*`.** `print("=" * 30)` prints 30 equals signs in a row.
2. **Join text and a number with `str()`.** `print("TOTAL: " + str(total))` prints `TOTAL: 371` when `total` is 371. Without `str()` Python gives an error, because it cannot add text and a number.
3. **Whole-number division `//` and remainder `%`.** `6 // 5` is `1` (6 divided by 5, whole part only). `6 % 5` is `1` (what is left over). The discount and the loyalty card use these.

1 The Banner **2 marks**

1. Make a variable that holds 30 equals signs: `"=" * 30`. You will print it three times in the receipt. (The starter already does this for you.)
2. Print that variable.
3. Print `CHAI CORNER`.
4. Print `FC College, Lahore`.
5. Print the variable again.

Expected output of Part 1:

```
==============================
CHAI CORNER
FC College, Lahore
==============================
```

2 The Cashier Block **3 marks**

1. Print `Cashier:` and then the name stored in `CASHIER`. Join them with `+`.
2. Print `Till:` and then the last three digits of `ROLL`. Take them with a slice: `ROLL[6:9]`. Do not turn it into a number. A roll number like `271050042` must show `Till: 042`, with the zero.
3. Cut the roll number into three pieces of 3 digits each: `ROLL[0:3]`, `ROLL[3:6]`, `ROLL[6:9]`. Turn each piece into a number with `int()`. Add the three numbers. For `271050784` that is $271 + 050 + 784 = 1105$.
4. Print `Receipt no:` and then that number. Remember `str()`: you cannot join text and a number without it.
5. Make a variable that holds 30 minus signs: `"-" * 30`. Print it.

Expected output of Part 2 (for Sara Ahmed, roll number 271050784):

```
Cashier: Sara Ahmed
Till: 784
Receipt no: 1105
------------------------------
```

3 The Order **4 marks**

Do these steps for item 1. Then do the same steps again for item 2, and again for item 3.

1. Ask for the name: `name1 = input("Item 1 name:  ")`.
2. Ask for the price. What `input()` gives you is text, so turn it into a number: `price1 = int(input("Item 1 price:  "))`.
3. Ask for the quantity in the same way: `qty1 = int(input("Item 1 quantity:  "))`.
4. Work out the amount: price times quantity. Store it in a variable, you need it again in Part 4.
5. Print one line: the name, a colon, the quantity, `x`, the price, `=`, the amount. One space between each piece. Use `str()` for every number.

After the three items, print the line of 30 minus signs. A quantity of 0 is allowed; then the amount is 0.

Expected output of Part 3 (for 2 Masala Chai at 60, 3 Samosa at 40, 1 Chicken Roll at 150):

```
Masala Chai: 2 x 60 = 120
Samosa: 3 x 40 = 120
Chicken Roll: 1 x 150 = 150
------------------------------
```

The text inside `input("...")` is called a *prompt*. It is what the program shows when it asks you to type. The prompt text is not marked; you may change it. But keep it inside `input(...)`. Do not show it with `print()`, because the test file thinks every `print()` is a line of the receipt.

4 Totals and Payment **3 marks**

1. Add the three amounts. Call the sum `subtotal`. Print `Subtotal:` and the subtotal.
2. Every customer gets 5% off. The discount is `subtotal * 5 // 100`. Print `Discount 5%:` and the discount.
3. The total is subtotal minus discount. Print `TOTAL:` and the total.
4. Ask for the cash: `cash = int(input("Cash received: "))`. Print `Cash:` and the cash.
5. The change is cash minus total. Print `Change:` and the change. (The cash is always enough, so the change is never negative.)
6. Print the line of 30 minus signs.

Expected output of Part 4 (for the order above and 500 rupees cash):

```
Subtotal: 390
Discount 5%: 19
TOTAL: 371
Cash: 500
Change: 129
------------------------------
```

5 The Loyalty Card **3 marks**

How the card works. Chai Corner gives every customer a small paper card with 5 empty boxes. Each time the customer buys one item, the shop puts one stamp in one box. When all 5 boxes have a stamp, the card is full: the customer gets one free chai, the full card is thrown away, and the customer gets a new card with 5 empty boxes.

Example. A customer buys 6 items in one visit. The first 5 stamps fill the card. That is one free chai. The 6th stamp goes on a new card. So the receipt shows a card with 1 stamp and 4 empty boxes, and says 1 free chai was earned.

Items bought	Full cards (free chais)	Stamps on the new card	Card shown
3	0	3	`[*][*][*][ ][ ]`
5	1	0	`[ ][ ][ ][ ][ ]`
6	1	1	`[*][ ][ ][ ][ ]`
12	2	2	`[*][*][ ][ ][ ]`

Now write the code, one small step at a time:

1. How many items did the customer buy in total? Add the three quantities. Put the sum in a variable.
2. How many full cards is that? Think: 12 items, 5 stamps per card, how many complete cards? You have an operator that divides and throws away the remainder. Put the answer in a variable.
3. How many stamps are left over for the new card? For 12 items it is 2. You have an operator that gives exactly the leftover part of a division. Put the answer in a variable.

4. Now draw the card. A full box looks like `[*]` and an empty box looks like `[ ]`. The card always has 5 boxes: first the full boxes, then the empty ones. You know how to repeat a piece of text. You know how many boxes are full. So how many are empty?
5. Print `Stamps: ` and then the card.
6. Print `Free chais earned: ` and then the number of full cards. Remember `str()`.

Try each step in the Shell first with a real number, for example 12. When the numbers in the Shell match the table, put the same expressions in your program.

Expected output of Part 5 (for the order above: $2 + 3 + 1 = 6$ items):

```
Stamps: [*][ ][ ][ ][ ]
Free chais earned: 1
```

6 Four Questions and the Footer **3 marks**

You need Lecture 4 (Thursday) for this part. It takes about 20 minutes after that class. Python gives you `True` or `False` when you compare two things, for example `cash == total`.

1. Combo: `True` if all three quantities are 1 or more. Hint: three comparisons joined with `and`. Store the answer in a variable. Print `Combo: ` and the variable.
2. Big order: `True` if the subtotal (before the discount) is 500 or more. Print `Big order: ` and the answer.
3. Chai on order: `True` if at least one of the three names is exactly `Masala Chai`. Use `or` with three comparisons. Capital letters matter: `masala chai` is not the same. Print `Chai on order: ` and the answer.
4. Exact cash: `True` if the cash is equal to the total. Use `==` (two equals signs). Print `Exact cash: ` and the answer.
5. Print the line of 30 equals signs.
6. Print `Thank you, come again!`

Careful: `True`/`False` is not text. `"Combo: " + combo` gives an error. Write `"Combo: " + str(combo)`.

Expected output of Part 6 (for the order above):

```
Combo: True
Big order: False
Chai on order: True
Exact cash: False
==============================
Thank you, come again!
```

Marking and Hand-in

Parts 1 to 6 are worth 18 marks, as shown next to each part. We run your program with several orders, not only the three in the test file. 1 mark for a comment on every block and using only Lecture 1 to 4 tools. 1 mark for a complete cover sheet with dev notes. Total: 20 marks. In every assignment of this course, a few students are asked in the lab to explain their code to a TA for two minutes. Your comments will help you do that.

- Save your file as `a0.py`. **Do not change the file name. The autograder will not find your file.**
- **Run `test_a0_student.py` before you submit. All checks must say PASS.** Remember to put the real `input(...)` lines back.
- Submit on Moodle. Late submission: minus 10% for each day, up to 3 days.

Common Mistakes

- `TypeError: can only concatenate str`: you added a number or `True`/`False` to text. Put it inside `str()`.

- **The line is wrong only in the spaces**: check the space after the colon, and the spaces around `x` and `=`.
- **The discount shows 19.5**: you used `/`. Use `//`.
- **Chai on order is always False**: `name1 == "Masala Chai" or "Masala Chai"` is wrong. Every side of `or` needs its own comparison: `name1 == "Masala Chai" or name2 == "Masala Chai" or ...`
- **The test file says the program asked for too many inputs**: `CASHIER` and `ROLL` must be written directly in the file, not asked with `input()`.