

Computing

Lesson 6: Create a Program

Programming Part 4: Subroutines

Ben Garside



Task 1 - Testing table

Complete the testing table on the next slide to ensure that this (oaknat.uk/comp-ks4-guessnumber) program works correctly. Remember to use erroneous, boundary and normal data for your tests. If the program produces an error then work out how to fix it.

Note: the code is provided below for you to copy and paste into your development environment if required.

```
to_guess = 4
number = 0
not_guessed = to_guess != number

while not_guessed:
    print("Guess a number between 1 and 10")
    number = int(input())
    if number < 1 or number > 10:
        print("Number out of range")
    elif number == to_guess:
        not_guessed = False
print("You got it!")
```



Task 1 - Testing table

Test number	Test description	Input (if required)	Expected output	Actual output	If the test was unsuccessful. How was it fixed?



Task 2 - Copington Adventure Theme Park

Copington Adventure Theme Park has just purchased an automated ticketing system and would like you to program the interface for customers to use. The ticketing system should ask the customer various questions about their visit requirements before providing a total charge, asking for payment and issuing a ticket.

Criteria

- Provide a welcome message
- Display the entrance ticket prices
- Ask how many adult tickets are required
- Ask how many child tickets are required
- Ask how many senior citizen tickets are required
- Ask how many of the visitors would like wristbands for the rides
- Ask for the lead Booker surname (for the ticket)
- Ask if they require a parking pass for the car park
- Display the total cost
- Ask for payment (the machine only accepts £10 and £20 notes, each note entered will need to be counted)
- Display change (if any)
- Print a ticket (displaying lead Booker surname, tickets purchased, wristbands purchased, today's date*)
- Print a car parking pass (if requested)
- Use data validation techniques to avoid runtime errors through incorrect data entry.
- Thank the customer for their purchase



Task 2 - Charges

Copington Theme Park Charges

Entrance Ticket Type

- Adult £20
- Child £12
- Senior Citizen £11

Wristband

- All £20

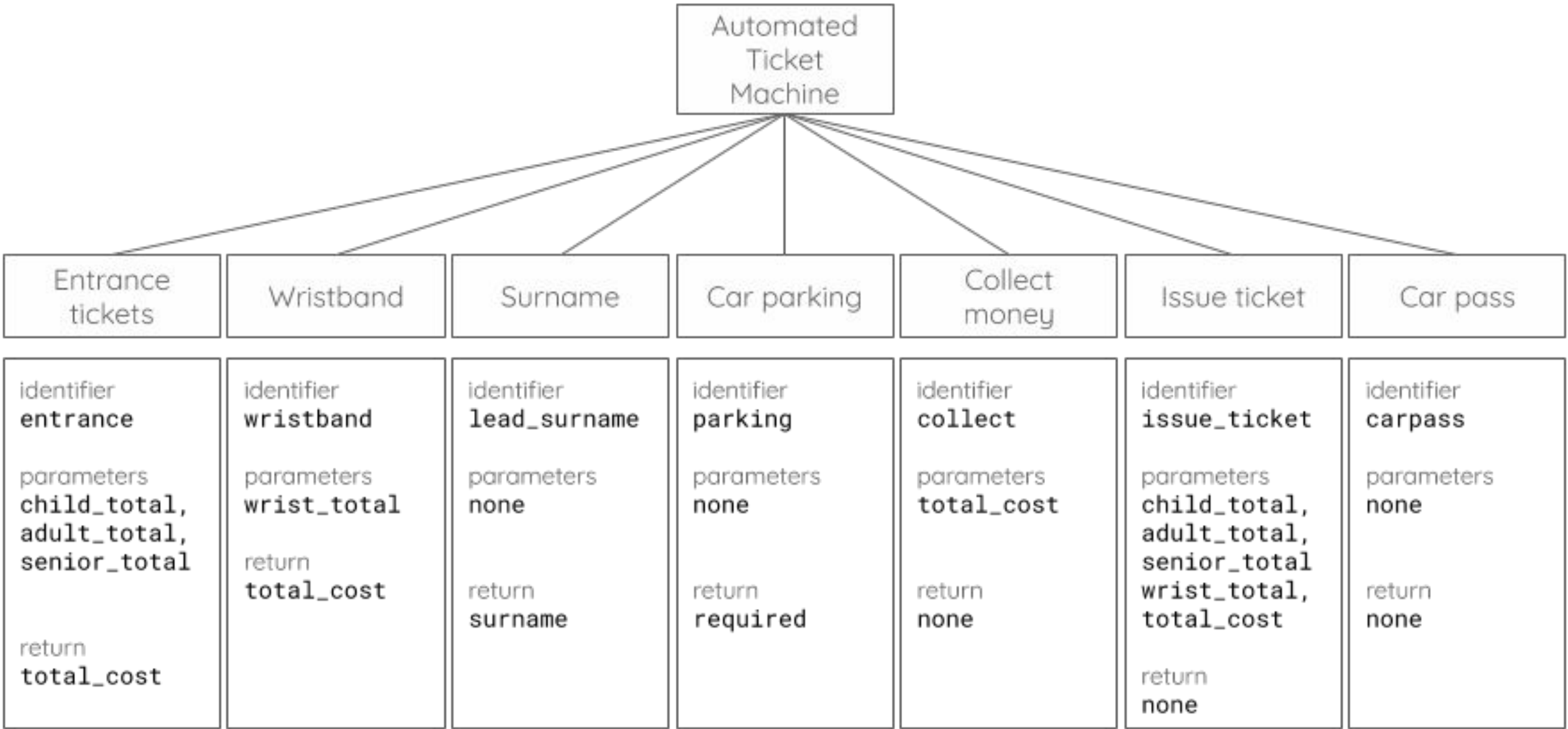
Parking

- Free (car pass must be displayed)



Task 2 - Planning and designing

A structure chart has already been created for the automated ticket machine to help you begin your design.



Task 2 - Design

Use pseudocode (oaknat.uk/comp-ks4-pseudocode) or a flowchart to design the algorithm for the program. Think carefully about what will be needed in the main program and in each subroutine.

Use the following slide to document/create your flowchart or pseudocode.

TIP: Use the structure chart in the previous slide to help you with variable names and subroutine structure.



Task 2 - Pseudocode / flowchart

If writing pseudocode, use the space below. If creating a flowchart, delete the box below and paste your flowchart on this side.



Task 2 - Create

Now you can started creating your coded solution.
Don't forget to **iteratively test** your program as you create it. This should help you to quickly find and fix any syntax, logic or runtime errors.



Task 2 - Testing

Perform final testing by completing the test table to ensure that all aspects of your program are functioning correctly.

Test number	Test description	Input (if required)	Expected output	Actual output	If the test was unsuccessful. How was it fixed?

