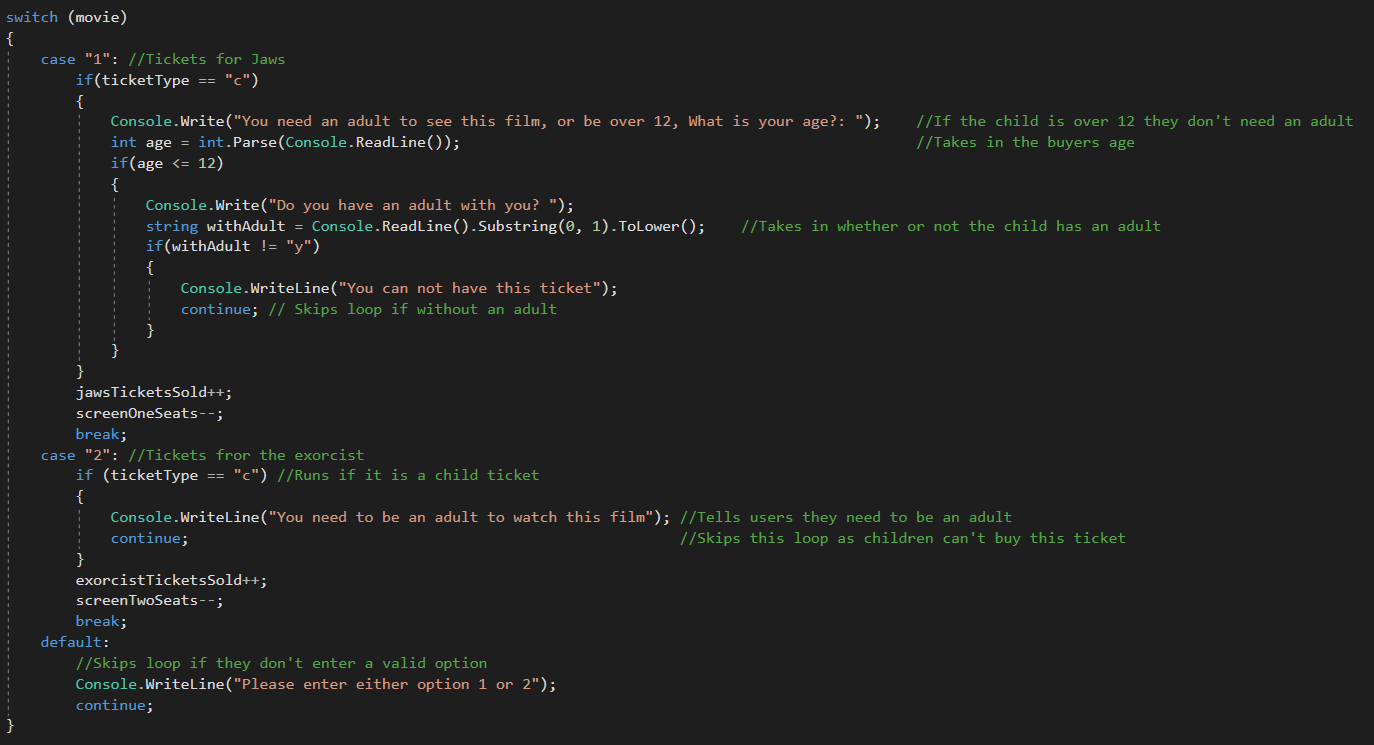
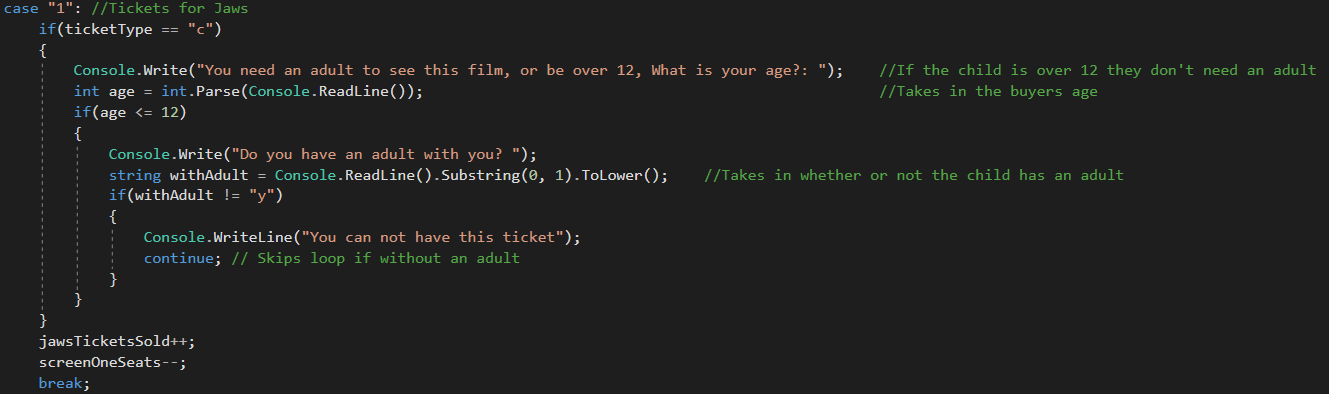
# Task 2

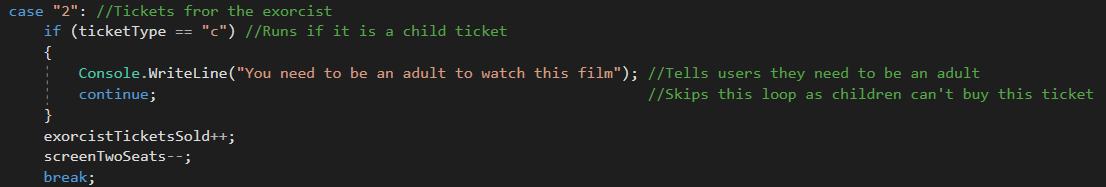
The ticket sales machine need to make sure that the age ratings were enforced, such as no under 18s going to see the Exorcist, and only teens and under 12-year olds with adults being able to see Jaws. To do this I used selective control structures, such as switches and If statements.



First, I asked the user what film they want to watch and what kind of ticket they want to buy. They choose from “Jaws” or “The Exorcist”, and then after they choose a film, they choose from adult, child, and student tickets. It first goes to a switch, which is used for deciding the movie. The user would say 1 for “Jaws”, or 2 for “The Exorcist”.



If the buyer chooses Jaws, it will check what ticket type has been chosen. If a child ticket has been chosen, it goes to an if statement to check that the buyer is allowed to go to the film. It first tells the user that they need to have an adult or be over 12 to watch the film, and then asks for the user to enter their age. If their age is under 12, it will ask them whether or not they have an adult with them. If the answer isn’t a yes, it will tell the user that they’re not allowed to buy this ticket and will then use the continue keyword to skip the rest of the loop, which in turn skips the code that sells the tickets. If the user entered an age over 12, or said that they have an adult with them, then it will run the code that purchases the tickets



This is the code for a child trying to buy a ticket to “The Exorcist”. It just tells the user that they are not allowed to buy this ticket, and then skips the loop with continue. This also skips the code that sells the tickets.

## Trace tables

### Jaws Adult Ticket

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Line | movieChoice | ticketType | adultTicketsSold | screenOneSeats | Output | Comments |
| 36 | 1 |  | 0 | 15 |  | User inputs 1 to select Jaws |
| 71 | 1 | A | 0 | 15 |  | User inputs a to select adult ticket |
| 102 | 1 | A | 0 | 14 |  | Screen one seats decremented by one |
| 136 | 1 | A | 1 | 14 |  | Adult tickets sold incremented by one |
| 134 | 1 | A | 1 | 14 | You have bought this ticket! | The program tells the user they’ve bought the ticket |
| 135 | 1 | A | 1 | 14 | Seats left in Screen 1 is 14 | The program tells you how many seats are left in the screen |

### Jaws Child Ticket(under 12, no adult)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Line | movieChoice | ticketType | Age | withAdult | childTicketsSold | Output | Comments |
| 36 | 1 |  |  |  | 0 |  | User inputs 1 to select Jaws |
| 71 | 1 | C |  |  | 0 |  | User inputs c to select child ticket |
| 91 | 1 | C | 10 |  | 0 |  | User inputs age less than 12 |
| 94 | 1 | C | 10 | No | 0 |  | User says they don’t have an adult with them |
| 97 | 1 | C | 10 | No | 0 | You cannot have this ticket | The program tells the user they’re not allowed to buy this ticket |

### Jaws Child Ticket(under 12, with adult)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Line | movieChoice | ticketType | Age | withAdult | screenOneSeats | childTicketsSold | Output | Comments |
| 36 | 1 |  |  |  | 15 | 0 |  | User inputs 1 to select Jaws |
| 71 | 1 | C |  |  | 15 | 0 |  | User inputs c to select child ticket |
| 91 | 1 | C | 10 |  | 15 | 0 |  | User inputs age less than 12 |
| 94 | 1 | C | 10 | Yes | 15 | 0 |  | User says they have an adult with them |
| 103 | 1 | C | 10 | Yes | 14 | 0 |  | The program tells the user they’re not allowed to buy this ticket |
| 142 | 1 | C | 15 | Yes | 14 | 1 | You have bought this ticket! | Child tickets sold increments by one |
| 144 | 1 | C | 15 | Yes | 14 | 1 | Seats left in Screen 1 is 14 | The program tells you how many seats are left in the screen |

### Jaws Child Ticket(Over 12)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Line | movieChoice | ticketType | Age | screenOneSeats | childTicketsSold | Output | Comments |
| 36 | 1 |  |  | 15 | 0 |  | User inputs 1 to select Jaws |
| 71 | 1 | C |  | 15 | 0 |  | User inputs c to select child ticket |
| 91 | 1 | C | 15 | 15 | 0 |  | User inputs age over 12 |
| 103 | 1 | C | 15 | 14 | 0 |  | Screen one seats decremented by one |
| 142 | 1 | C | 15 | 14 | 1 | You have bought this ticket! | Child tickets sold increments by one |
| 144 | 1 | C | 15 | 14 | 1 | Seats left in Screen 1 is 14 | The program tells you how many seats are left in the screen |

### The Exorcist Child Ticket

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Line | movieChoice | ticketType | childTicketsSold | Output | Comments |
| 36 | 2 |  | 0 |  | User inputs 2 to select The Exorcist |
| 72 | 2 | C | 0 |  | User inputs c to select child ticket |
| 116 | 2 | C | 0 | You need to be an adult to watch this film | The program tells the user that they cannot buy the ticket |

### The Exorcist adult ticket

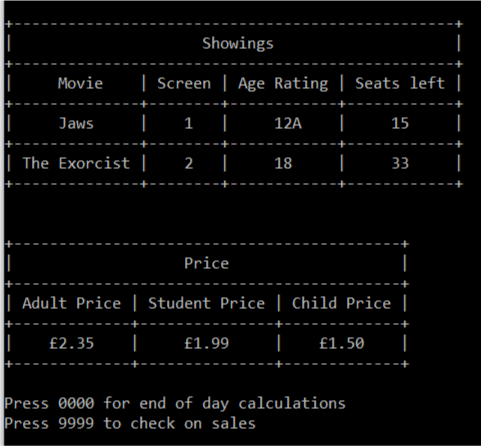
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Line | movieChoice | ticketType | adultTicketsSold | screenTwoSeats | Output | Comments |
| 36 | 2 |  | 0 | 33 |  | User inputs 2 to select The Exorcist |
| 72 | 2 | A | 0 | 33 |  | User inputs a to select child ticket |
| 125 | 2 | A | 0 | 32 |  | The program decrements by one |
| 137 | 2 | A | 1 | 32 | You have bought this ticket! | Adult tickets sold increments by one |
| 139 | 2 | A | 1 | 32 | Seats left in Screen 2 is 32 | The program tells you how many seats are left in the screen |

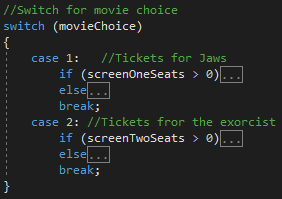
# Task 3

Currently, the only way to add new films is to hardcode the extra options. This would be done by adding more case statements to the switch that deals with the choice of film. Thankfully if there is another 12a rated film, a lot of the code can be reused from Jaws, cutting down on development time.

There are problems with changing the price after some tickets have already been sold, as it would complicate the end of day calculations. My program currently calculates the end of day price by multiplying the amount of the type of ticket sold and the price of that ticket. If we were to change the price of the ticket during the day, it would require more complex calculations. There is no reason why they couldn’t change the price of tickets at the beginning of the day before any tickets have been sold, as this would cause no complications at the end of the day.

If the ticket prices are dependent on the time of day, such as early bird tickets for films being shown in the morning, then the time of day could be checked and then the prices could be set. To do this, I would implement different ticket prices, such as morningAdult, afternoonAdult, eveningAdult. This means I could then times the morning adult ticket with the morning prices, etc, etc.



At the bottom of the welcome screen, it has options, such as press 9999 to check on sales. This option runs the end of day method and then continues the loop. this allows staff such as mangers to check the sales for the day without finishing for the day. To add more films, another option could be added, such as typing admin for an admin backend, that allowed the user to change the films being shown, and the relevant information about them, such as the age rating and the screen it’s being shown in. This admin backend could also allow the user to change the price of the tickets.   
The problem with this method is that the programmer is required to manually enter the new options into the source code. The screenshot to the left shows the source code for the movie choice. Depending which number the user inputs, the switch will run code for the appropriate film. This means that if a new film was to be entered, it would have to involve manual coding for both the display of what films are available as well as the code to buy tickets for it.