### 

|  |  |  |  |
| --- | --- | --- | --- |
| **Programme Name** | Undergraduate Computing Suite | | |
| **Module Title** | Introduction to Programming | | |
| **Module Code** | MCOMD1PRG | | |
| **Module Start Date** | 24/9/18 | | |
| **Module Level** | 4 | **Assessment Type(s)** | Assignment |
| **Word Length / Duration** | 2000 | **% contribution to module mark** | 50 |
| **Submission Date** | 07/01/19 | **Submission Location** | Blackboard |
| **Feedback Date** | With results |  |  |

**General advice:**

* You are required to back up your work regularly onto your N: drive and on removable or cloud storage devices
* Always check the date-stamp on your files before submission, and submit the latest versions
* ***You must submit your work using the software versions currently on the University’s network***

|  |
| --- |
| **Learning Outcomes tested by this assignment:** |
|  |
| 1. Demonstrate a basic understanding of initial computer programming constructs of syntax, sequence, selection and iteration, input and output and data types and variables. 2. Utilise effectively basic editing software and an integrated development environment. 3. Interpret simple specifications and design and implement well-formed solutions to these. |
|  |

**Scenario**

The Sarre International Cinema Corporation has a single cinema located on the campus of the University of Sarre[[1]](#footnote-1). It shows modern blockbuster films to local children, adults and students. They need a Point of Sale (POS) system to manage the sale of tickets over the course of a day. There are two screens with each showing films once per day. Each screen has a limited number of seats.

The current films are as shown:

|  |  |  |  |
| --- | --- | --- | --- |
| Film | Certificate | Seats | Screen |
| Jaws[[2]](#footnote-2) | 12A | 15 | 1 |
| The Exorcist[[3]](#footnote-3) | 18 | 33 | 2 |

The system should not be able to sell more tickets for a screen than there are seats in its capacity.

The prices of entry are:

|  |  |
| --- | --- |
| Type | Price |
| Child | 1.50 |
| Adult | 2.35 |
| Student | 1.99 |

However, a child should not enter the cinema without at least one accompanying adult.

The system also has to calculate the VAT amount that the cinema has to pay at the end of the day. This is to be done as follows:

1. Total-Ticket-Price = Num-Adult-Tickets-Sold \* Adult-Ticket-Price +

Num-Child-Tickets-Sold \* Child-Ticket-Price +

Num-Student-Tickets-Sold \* Student-Ticket-Price

1. Pre-VAT-Price-Total = roundup(Total-Ticket-Price / 1.2)
2. VAT-amount-Total = TotalTicketPrice – Pre-VAT-Price-Total

When the VAT rate is 20%.

A report at the end of the day should show the following information:

Hello to Jason Isaacs

Number of Child Seats Sold: 4

Number of Adult Seats Sold: 7

Number of Student Seats Sold: 5

Showing in Cinema 1: Jaws certificate 12A

Number of Seats in Cinema 1: 11

Showing in Cinema 2: The Exorcist certificate 18

Number of Seats in Cinema 2: 5

Days Takings £32.40

Takings for cinema £27.00

VAT to Pay £5.40

Press a key to quit.

## Task 1

Your task is to write a C# program that will implement a system as described in the scenario above. You must use a Console App (.NET Framework) project to do this.

### Submission:

You are to submit the Program.cs file ONLY onto the Blackboard Submission tool in the appropriate area on Blackboard.

## Task 2

Explain carefully using a section of your code how you have prevented children purchasing tickets unaccompanied by an adult. Show different traces through your code for different input by the user to demonstrate that it handles these appropriately.

### Submission:

You are to submit this as part of a document to Turnitin on the appropriate area on Blackboard.

## Task 3

The Director of the Cinema Corporation has suggested that they should be able to have two performances per day at the cinema and also to change the films being shown in the screens for a second performance of the day. He also has suggested that the price of the different tickets should be updatable using the system after the some tickets have been sold. Can you identify potential problems with this suggestion in terms of the computer program and the calculations that are meant to be made by the system? Describe the issue as if to the Director who is not a computer expert[[4]](#footnote-4).

Research and describe briefly a solution that could overcome the problem. Please note you should not implement this solution, but simply need to describe it.

### Submission

You are to submit this report as part of a document to Turnitin on the appropriate area on Blackboard.

**Introduction to Programming**

ASSIGNMENT 1: Sarre International Cinema Corporation

Date Set: 12 November 2018

Course Tutor: David Bennett

Feedback Sheet

Student’s Name: Final Grade:

Max Mark Awarded

Implementation

Sell Ticket 20

Show seat availability 10

End of Day Calculations 10

Use of variables 10

Use of control structures 10

Attractive and Easy to Use 5

Discussion

Child entry requiring adult accompaniment 10

Issues with Film and Price change 15

Suggested method for solution 10

100

**Comments:**

# Mark Scheme

**The mark scheme below is indicative and marks may be awarded or deducted outside the precise scope of what is listed for particularly good or particularly poor work in an area.**

### Sell Ticket (20)

0-7 No Sales of tickets possible, or sales of tickets severely limited

8-11 Sales of different type of ticket possible, no tracking over course of day, possible slip in calculations, possible to buy more tickets than available

12-15 Sales of different type of tickets possible, tracking over course of day

16-20 Ticket sales recorded and rules for children imposed successfully

### Show seat availability (10)

0-3 Availability not shown

4-5 Number of tickets sold, but not availability

6-7 Ticket availability for one screen shown, but not both screens

8-10 Ticket availability for each screen shown

### End of Day Calculations (10)

0-3 No end of day calculations

4-5 One or two of end of day info correct

6-7 All values present, may have a slip in one calculation (e.g. VAT)

8-10 All values present and correct

### Use of variables (10)

0-3 No use of variables

4-5 variables used, but wrong data types and poorly named more often than not

6-7 variables used, good data types good naming, not necessarily consistent

8-10 variables used with good data types and good and consistent naming, declared in a suitable place

### Use of control structures (10)

0-3 No use of control structures (other than sequence)

4-5 Basic use of conditional OR iterative statements

6-7 Good use of conditional AND iterative statements – program loops for whole day, good layout of code

8-10 Good use of conditions and iterative statements, appropriate selection, good layout of code

### Attractive and Easy to Use (5)

0-1 Not usable

2 Basic prompts

3 Suitable prompts in many places and suitable layout on screen

4-5 Suitable prompts for main flow and error and good layout on the screen to make information easily readable.

### Child entry requiring adult accompaniment (10)

0-3 No Description, or very weak and unclear, no, or incorrect code trace with poor values

4-5 Basic description, doesn’t use correct terminology and some errors, not fully clear, code trace for one or two values used, selection of values could have been better

6-7 Good description, starting to use correct terminology, generally clear, code sample described appropriately, suitable values used for code traces

8-10 Excellent description, with suitable values for code trace, good use of terminology throughout

### Issues with Film and Price change (15)

0-5 No or inappropriate issues raised

6-9 One basic issue raised, description weak, but understandable – not necessarily by intended audience

10-13 Several appropriate issues raised, description acceptable, but not necessarily for intended audience

14-15 Good description, several appropriate issues raised and understandable by intended audience

### Suggested method for solution (10)

0-3 No method or inappropriate method described

4-5 Method provided, but not optimal (given teaching so far in module)

6-7 Good method provided, but not well described

8-10 Good method well described.

# General Advice – Backing up your code

Every year we see students who have lost all their work just before it is due to be handed in owing to computer faults or their own mistakes. Usually this is entirely avoidable and at most only a few hours of work should have been lost. It usually happens because students have not taken adequate back-ups of their work.

Our advice is that AT LEAST ONCE per day that you should make a copy of the work you have done so far and make a backup of this on at least two separate devices – one on the computer you are working on and one on a cloud based area such as the OneDrive provided by the University OR the University N drive. This can also be supplemented with a backup to a further device, like an external hard drive, a large capacity pen drive or a different cloud service, such as Google Drive etc..

To make a sensible backup make a copy of the WHOLE folder you have been working in, rename the copied folder to include the date and time in reverse format and copy this to the multiple location. We would advise keeping a good number of these and as a minimum 3 days’ worth.

Reverse date format is of the form YYYYMMDD, so that the year is followed by the month then the day. This allows easy sorting of the folders to make it easier to find a particular folder.

If you decide you have gone wrong, and want to start again, again we would advise KEEPING all the old backups you have in case it turns out that you do not progress at all well in the new version – at least then you have the option to hand in an older version and get some marks for what you did previously, even if it’s not maximal.

# General Advice – Plagiarism

The biggest reason for plagiarism in assignments is due to leaving an assignment until the last minute. To counter this, make sure you know *what* you have to do (not necessarily the same as *how* to do it) as soon as the assignment is set. You will almost certainly have to do some background reading and programming practice in order to complete the assignment. If you are unsure of what is required, speak to your tutor in the lab.

You may discuss aspects of this assignment with other students, but do not work closely with others while at a computer and working on the assignment. Certainly don’t sit next to each other typing, more or less, the same code. This is an *individual* assignment where we are testing your *individual* skills. It’s really obvious for us to detect when students have worked together and will almost certainly result with everyone involved scoring zero (if this is the first instance of plagiarism).

Never give another student access to your code, regardless of how desperate they are. Tell those students to seek help from their tutor. If someone is really pressuring you, contact your tutor immediately. Giving away your hard work will result in both of you scoring zero, and that student never learning.

This advice is not an exhaustive list on what constitutes plagiarism; see your course handbook for more details.

1. Their adverts uses the tag line, ‘When you visit Sarre, you’ve seen it!’. [↑](#footnote-ref-1)
2. It’s a film about a shark. This is not Jaws 2– its just the superscript for the footnote! [↑](#footnote-ref-2)
3. It’s a film not about a shark. This is not The Exorcist 3. See above. [↑](#footnote-ref-3)
4. This is NOT about there may be upset customers who are expecting to see a different film – you can assume that all the people who have bought tickets for a particular screen will be excited to see a new film. Customer already with tickets would not have to pay any more if the price changed. [↑](#footnote-ref-4)