Final Exam

Started: Apr 11 at 7:04p.m.

Quiz Instructions

Please make sure to read the following instructions carefully before beginning the test.

- **This is an individual test, you are not to share your answers with another individual**
- The test is out of 46 marks
- The point value of each question is shown at the top right of the question box
- You have 3 hours to complete this test
- If you are late logging on to the test, the time will not be extended
- Read each question carefully and make sure you answer all parts of each question
- You have access to the Visual Studio IDE
- You have access to the Visual Studio Documentation, but no other Internet resources
  - https://docs.microsoft.com/en-us/dotnet/csharp/ (https://docs.microsoft.com/en-us/dotnet/csharp/)
- Syntax is important and your answers must compile for full marks
- ## When submitting code, please be sure to zip and upload your ENTIRE project (if you fail to upload all files, a mark of zero will be given)

Question 1 14 pts

Assume that a supermarket has two checkout lines, each leading to a cashier. You are to write a console program that simulates customers lining up at the shortest line (least customers) to pay for their groceries.

Every minute, there is a 50% chance that a new customer will arrive at the checkout lines. Once a customer reaches the front of the line, it takes between 1 to 10 minutes to process them. Add comment with "Checkout lane Exam 23" to top of any code file.

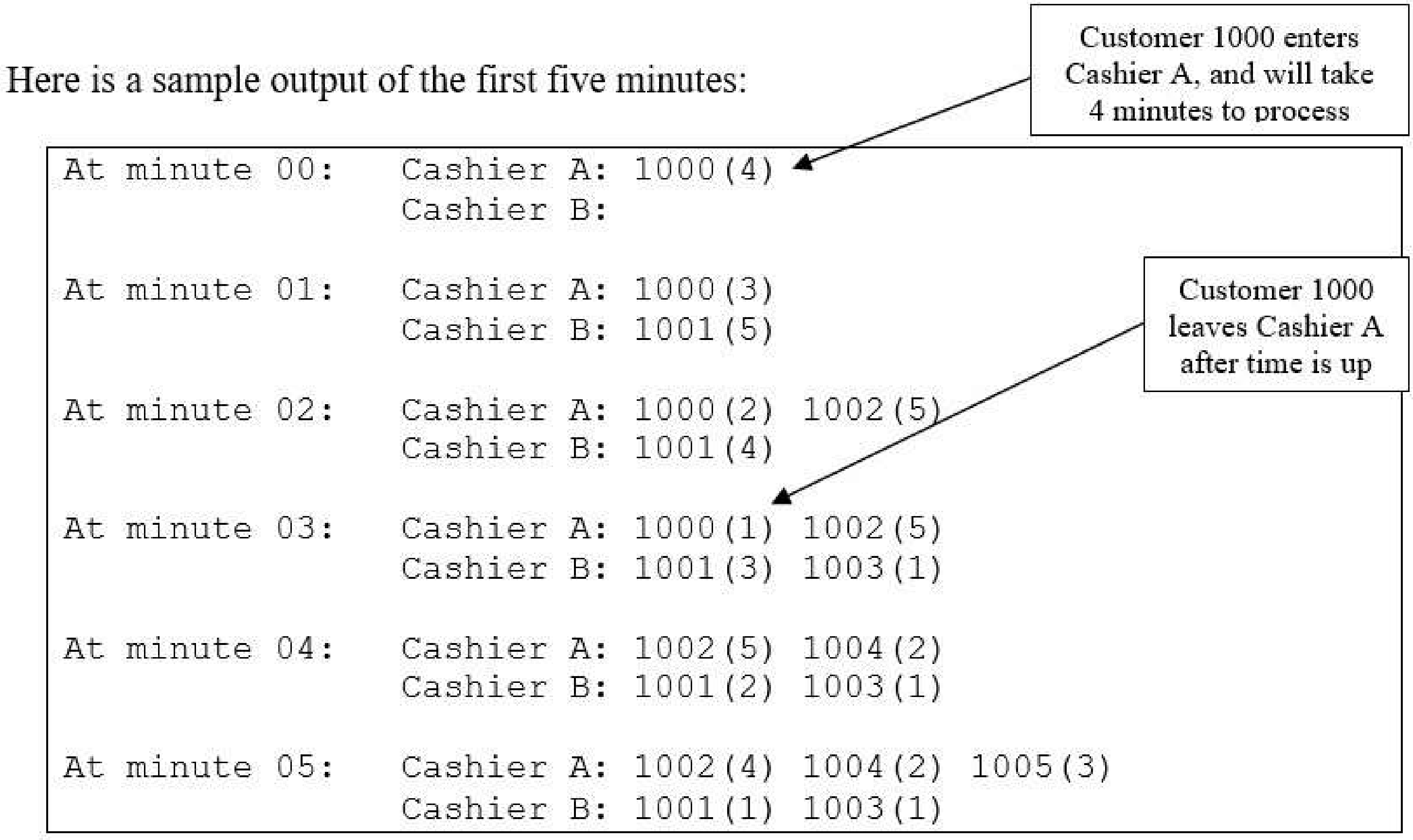
The simulation is to run for 30 minutes (not real time). At the end of every simulated minute, display all of the customers that are lined up in both queues as well as their remaining wait times.

You are to make use of the generic ***List<T>*** collection. Below is the ***Customer*** class:

```
public class Customer
{
  public int Number { get; private set; }
  public int TimeLeft {get; set;}
  private static Random r = new Random();
  private static int count = 0;

  public Customer()
  {
    Number = 1000 + count++;            //Unique customer ID
    TimeLeft = r.Next(10) + 1;          //Wait 1 to 10 minutes
  }

  public override string ToString()
  {
    return string.Format($"{Number}({TimeLeft})");
  }
}
```



Here is a sample output of the first five minutes:

```
At minute 00:   Cashier A: 1000(4)
                Cashier B:

At minute 01:   Cashier A: 1000(3)
                Cashier B: 1001(5)

At minute 02:   Cashier A: 1000(2)  1002(5)
                Cashier B: 1001(4)

At minute 03:   Cashier A: 1000(1)  1002(5)
                Cashier B: 1001(3)  1003(1)

At minute 04:   Cashier A: 1002(5)  1004(2)
                Cashier B: 1001(2)  1003(1)

At minute 05:   Cashier A: 1002(4)  1004(2)  1005(3)
                Cashier B: 1001(1)  1003(1)
```

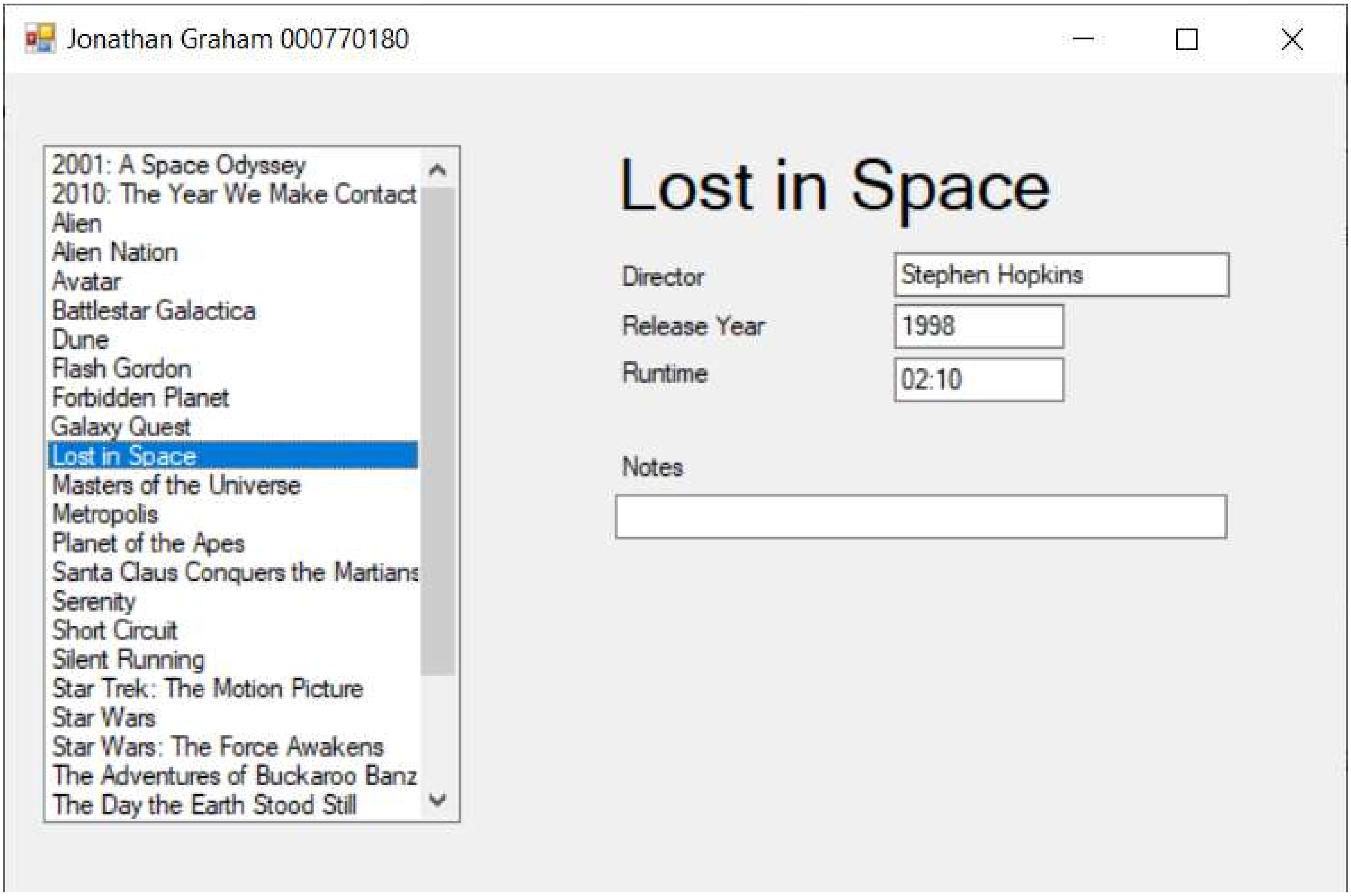
Customer 1000 enters Cashier A, and will take 4 minutes to process

Customer 1000 leaves Cashier A after time is up

Upload

Choose a File

Here is what your UI should look like for the following question:



Jonathan Graham 000770180

2001: A Space Odyssey
2010: The Year We Make Contact
Alien
Alien Nation
Avatar
Battlestar Galactica
Dune
Flash Gordon
Forbidden Planet
Galaxy Quest
Lost in Space
Masters of the Universe
Metropolis
Planet of the Apes
Santa Claus Conquers the Martians
Serenity
Short Circuit
Silent Running
Star Trek: The Motion Picture
Star Wars
Star Wars: The Force Awakens
The Adventures of Buckaroo Banz
The Day the Earth Stood Still

Lost in Space

Director: Stephen Hopkins
Release Year: 1998
Runtime: 02:10

Notes

Question 2 32 pts

For this question, you will need to create a small database on your **MS SQL Server Management Studio**, and then write a Windows Form (.NET Framework) application to access it. Do not worry; the script to create the database is already complete.

a. Start **MS SQL Server Management Studio** (or alternatively **MS Visual Studio** if you know how to run SQL scripts from inside there).
b. If using **MS SQL Server Management Studio**, select "**.\YourServerName**" as the server name, and **Connect**. Your server name may be "**SQLExpress**", but may be different. Use whichever server name you have been using.
c. Select the **New Query** button on the toolbar.

d. Copy and paste the SQL script file, **CreateSCIFI.sql (https://mycanvas.mohawkcollege.ca/assessment_questions/6324211/files/7870215/downl verifier=YfbTCFxjibhf325CR95AyAuimgaO66j1vUSkIG12&wrap=1)** **(https://mycanvas.mohawkcollege.ca/assessment_questions/6324211/files/7870215/downl verifier=YfbTCFxjibhf325CR95AyAuimgaO66j1vUSkIG12&download_frd=1)** , into the query window.
e. Click the **Execute** button on the toolbar.
f. **COMP10204_FINAL** will be the name of the database created.
g. Feel free to explore the database either here or in **MS Visual Studio**.

Using the image above as a template, create a Windows Form (.NET Framework) app that does the following:

1. Has your name and student number on the FORM TITLE.
2. Populates a listbox with all the movie titles from the database listed in descending order by release year.
3. When an item is selected from the listbox. Display it's Title, Director, Release Year, and Runtime.
4. The runtime in the database is stored in minutes. You must display it as HH:MM i.e., 70 minutes = 1 hour and 10 minutes => 01:10. Both hours and minutes should be zero padded (i.e. display "01" instead of "1")
5. Run a database query on the table "Ship" to retrieve all the spaceships from this film.
6. In the notes field, display the phrase "In this film we saw:" followed by a comma separated list of spaceships that appeared in this film. For example:

   ```
   In this film we saw:  Jupiter II, Proteus
   ```

7. If there are no matching records in the Ship table. Leave the notes field blank.

You may connect to the database in any manner you choose, be it using data aware controls, ADO.NET objects, or the Object Relational Designer.

You may use any SQL, LINQ, or Lambda Expressions as you choose.

Marks will be awarded for style, which includes proper naming conventions, appropriate use of form controls, best practices, efficient code, and a program that runs properly and displays the correct results. The only comment you need to provide is your name and student number. You may include additional comments, but they are not required,

4 Marks – GUI controls

6 Marks – Database Connection

6 Marks – Query

8 Marks – Display

4 Marks – Total runtime

4 Marks – Best Practices

-10% – Program does not compile

-10% – Program crashes when running

Upload

Choose a File

Quiz saved at 7:55pm Submit Quiz