

# **Department of Computer Science**

## **08101 Programming I**

### **Week 2 Laboratory 2007/2008**

#### **More Complex Programs**

We now know how to create simple C# programs and make them run. In this session you are going to develop the programs that have already been written, and do some more complicated data processing. Note that you will now have to invent your own code, as opposed to typing in what we provide.

#### **Making Decisions**

The program that we wrote last week simply runs to completion. It reads in some data, performs a calculation and then prints out the result. In this respect it is working just like a pocket calculator. However, most programs also need to have the ability to make decisions. This is one of the things which makes computers truly useful, in that your program can react to different types of data in an appropriate way.

#### **Cinema Entry Problem**

As an example of conditions in action we could consider a program which decides whether or not a particular person is allowed to go and see a film. If the age of the user and the film match up the program will print "Enjoy the film" otherwise it will print out "Access Denied - You are too young". Note that the system does not have any way of verifying the age, we are taking on trust that the user is not telling fibs! The first program we are going to write will perform the test for the film " Shoot'Em'Up " which has a minimum age requirement of 18 years. The program will have the following user interface:

```
The film you want to see is Shoot' Em' Up  
Enter your age: 21  
Enjoy the film
```

What the user types in is shown as underlined.

#### **Test Data**

Before you can be sure that your program will work you must make sure that you have some test data to use with it.



Before you go any further; perform the following:

1. Write down some test data (four values should be sufficient) to make sure that your program works (you should pick values which are close to the test boundary and others which are not).

Now that you have some test values you can write the code which does the testing. You will need to make use of the C# language construction which can make decisions. Otherwise known as the "if" construction.

## Writing a Program

Now you can write the program.



Before you go any further; perform the following:

2. Log on to a workstation using your username and password.
3. Start a command prompt.
4. Create a directory to store the files for this week:  
`md 081011ab02`
5. Use the `cd` command at the command prompt to move into your work directory.
6. Use the notepad program to create a program in the file `Cinema.cs` which works as required. A good starting point is to use the copy command to make a copy of your `Sums.cs` program from last week. This already has C# code which will print out messages and read numbers from the user.

## Completing the Solution

We can now go ahead and create the final version of the program. This will print a menu of films from which the user will choose the one they want to see. The user will then enter their age. The program will be used like this:

```
Welcome to our Multiplex
We are presently showing:
1. 3:10 to Yuma (15)
2. Atonement (15)
3. Bratz: the Movie (U)
4. Shoot'Em'Up (18)
5. Rush Hour 3 (12A)

Enter the number of the film you wish to see: 1
Enter your age: 12
Access denied - you are too young
```

Your program will have to read and store the number of the required film. It will then read and store the age. You must then use an appropriate arrangement of conditions to decide whether or not the customer can see the film.



Before you go any further; perform the following:

7. Devise a set of test cases which will allow you to test your program. There should be two test cases per film, one which works and one which does not.
8. Modify the `Cinema.cs` program so that it works as above and use your test data to prove this.

## Input Validation

At the moment the program is not perfect. It could be made to do stupid things. There are a number of ways in which a user could upset your program:

- enter an age value which is stupidly large
- enter an age value which is stupidly small
- enter a film value which is stupidly large
- enter a film value which is stupidly small

You will need to devise some "metadata" which describes the age and film values and what constitute valid values, as an example, the film number should never go below 1.



Before you go any further; perform the following:

1. Devise the valid values for the age and film values in your program.
2. You can add them as comments to the code as follows:  
`int FilmNo; // must not be less than 1 or bigger than 8`
3. Add the comments to your program and improve the code so that values outside the given ranges are rejected with an appropriate message.

## Double Glazing Calculations

In the lectures we have discussed a program to work out the area of glass and length of wood required to make a particular double glazed window. The customer has told you that the height must not be less than 0.5 metres or greater than 5.0 metres and the width must not be less than 0.5 metres or greater than 3.0 metres. Write some code which will print out appropriate messages if the values given by the user are out of range.

## A program of your own

For the remaining part of the laboratory you must create a program which works in a similar way to the cinema entry one. Pick one of the situations below and create a program with a sensible name which behaves in the required manner:

### Grading potatoes

A program to select the grade letter of a given potato. (Note that these grades are made up; you may be able to find proper potato weights on the web). The user will enter the weight of the potato and the grade will be displayed.

1. Less than 200 gms – grade X
2. Between 200 and 400 gms – grade A
3. Between 400 and 800 gms – grade B
4. Above 800 gms – grade Z

### Exam scores and degree classifications

A program to print out the degree classification (or not) of a candidate with a particular score. The user will enter the mark and the program would print out the classification:

1. Less than 35% - fail
2. Between 35% and 40% - compensatable
3. Between 40% and 50% - third class degree - III
4. Between 50% and 60% - lower second class degree – II(ii)
5. Between 60% and 70% - upper second class degree – II(i)
6. Above 70% - first class degree - I

### Call Centre

A program which allows the user to select a particular function for a dial in call centre. The user enters the command number and the program would then select the appropriate function. For our purposes it will just print out which option has been selected:

1. Press 1 to get an account balance
2. Press 2 to pay a bill
3. Press 3 to transfer money to another account
4. Press 4 to order a cheque book

5. Press 5 to order a statement
6. Press 6 to change your PIN
7. Press 7 to speak to an operator.

## **Boxing Weights**

A program to allow the user to determine the weight classification of a boxer. The user will enter the weight of the boxer in pounds and the program will print out the classification:

1. Heavyweight: Unlimited
2. Cruiserweight: Limit - 190 pounds
3. Light Heavyweight: Limit - 175 pounds
4. Super Middleweight: Limit - 168 pounds
5. Middleweight: Limit - 160 pounds
6. Junior Middleweight: Limit - 154 pounds
7. Welterweight: Limit - 147 pounds
8. Junior Welterweight: Limit - 140 pounds
9. Lightweight: Limit - 135 pounds
10. Junior Lightweight: Limit - 130 pounds
11. Featherweight: Limit - 126 pounds
12. Junior Featherweight: Limit - 122 pounds
13. Bantamweight: Limit - 118 pounds
14. Junior Bantamweight: Limit - 115 pounds
15. Flyweight: Limit - 112 pounds
16. Junior Flyweight: Limit - 108 pounds
17. Strawweight: Limit - 105 pounds

## **Vehicle Duty Rates**

A program to calculate the duty to be paid on a vehicle. The user enters the type of car, length of licence and emission band and the program prints out the cost of the license:

Bands	CO <sub>2</sub> Emission Figure (g/km)	Diesel Car TC 49		Petrol Car TC 48		Alternative Fuel Car TC 59	
		12 months rate £	6 months rate £	12 months rate £	6 months rate £	12 months rate £	6 months rate £
Band AA	Up to 120	80.00	44.00	70.00	38.50	60.00	33.00
Band A	121 – 150	110.00	60.50	100.00	55.00	90.00	49.50
Band B	151 – 165	130.00	71.50	120.00	66.00	110.00	60.50
Band C	166 – 185	150.00	82.50	140.00	77.00	130.00	71.50
Band D	Over 185	160.00	88.00	155.00	85.25	150.00	82.50