

M.Sc. in

COMMUNICATIONS ENGINEERING

SATELLITE AND MOBILE COMMUNICATIONS

COMMUNICATIONS NETWORKS

ELECTRONIC AND COMPUTER ENGINEERING

EMBEDDED SYSTEMS

INTERACTIVE DIGITAL MEDIA

also with

INDUSTRIAL STUDIES

Summer 2009

3 hrs

9:30 am to 12:30

**Introduction to Computing Programming Exercise**

**04 41480**

The programming exercise is designed for students who need to retake the coursework assignment of this module. The exercise is a time limited exercise of individual work completed in morning or a day. The test is conducted under open book examination conditions.

The assessment is made on the basis of program functionality and supplied code, which are specified in the detailed instructions issued on the day of the test.

Students may use any teaching materials previously supplied, available on webCT, the internet and personal notes.

Any material taken from elsewhere should be clearly identified in any report or program listing produced. Material from elsewhere, used without adaptation will not be assessed as the students own work.

If the source of material taken from elsewhere is not acknowledged, then the use of that material without acknowledgement will be regarded as plagiarism and the normal penalties applied.

It is permitted to adapt existing code for a solution with the limitations specified above. Students should remember that time spent searching for alternative sources is time not spent working on the solution.

Students may not confer during the test and must be escorted to and from the room and back should they need to avail themselves of washroom facilities.

Students are expected to use the computers and facilities provided by the University to complete this work.

The assignment instructions may not be removed from the laboratory.

Refreshments will, exceptionally, be permitted in the laboratory but must be placed in the designated area at the start of the session and consumed in that location, not alongside the computers or other equipment. Students may not leave the lab in order to purchase or collect refreshments.

Students may not leave the lab without a chaperone.

**Object Oriented Programming Resit Assignment**

You are required to produce a console based C# application to demonstrate the simple card game *Pairs*. A playing card has a *value* which runs from the ace, two, three … ten, jack, queen and king and a *suit* which is hearts, diamonds, spades and clubs. There are 52 cards in a complete deck. The application should produce suitable screen output to show the game being played. Once the game is started, it requires no user interaction and runs to a conclusion.

The game of Pairs is one of pure chance and its rules are as follows. It is a game for two players only. All of the cards in the shuffled deck are dealt alternately to each player so each player starts with a deck of 26 cards. One player is randomly chosen to play first and he lays down the top card from his deck face up. The second player lays his card face up on top of the first card. If the face value of the two cards is the same (for example a ten of clubs and a ten of hearts), the second player takes all the cards in the pile (in this case there are only two cards) and puts these cards on the bottom of his deck. The game continues with each player alternately laying down their top card and the player whose card matches the previously laid card in face value, ‘wins’ all of the cards in the pile and those cards go to the bottom of his deck. The player who ends up with no cards loses.

You will be expected to produce an object oriented solution to this assignment. **A** **procedural solution will result in you failing this assignment**. In order to allow you to structure your program, I have produced an outline template of the classes required. **You must use these classes in your program** and it will be up to you to decide on suitable fields and methods of these classes as well as the interactions between objects of these classes in order to implement the card game. You will also need a test class containing a main method. Thus your complete solution must comprise 5 classes in total.

class PlayingCard

{

}

class DeckofCards

{

}

class Player

{

}

class CardGame

{

}

Assessment

Assessment will be on the basis of an assessment of the functionality of your program and the code which you have produced. You are required to email me your source code in a single file (.cs). Please only email me your source code file and  **include your ID number in the source filename. You must do this at the end of the examination and no later than this**. Any special instructions for running your program must be included in the main text of the email but, since this is a console based application, this should not really be necessary. The proforma I will be using for assessing your program is included in this document. The marks allocated to each class are shown and reflect the amount of code required for that class.

Dr Mike Spann 13/7/09 [m.spann@bham.ac.uk](mailto:m.spann@bham.ac.uk)

***Introductory Module***

**04 41480**

**Object Oriented Programming**

|  |  |  |
| --- | --- | --- |
| **Code**   * PlayingCard class * Player class * DeckOfCards class * CardGame class * Test class   **Sub-total** |  | /5  /5  /10  /15  /5  /40 |
| **Program Functionality**  Some, full, extended |  | /30 |
| **Total** |  | /70 |