|  |  |  |  |
| --- | --- | --- | --- |
| **COMP1148 (2015/16)** | **Computer Programming (RESIT)** | **Header ID 227776** |  |
| **Course Leader Dr Chris Walshaw** | **Release Date Wednesday 18/05/2016** |  | **Deadline Date Sunday 17/07/2016** |
| This coursework should take an average student who is up-to-date with tutorial work approximately 50 hours   Feedback and grades are normally made available within 15 working days of the coursework deadline | | | |
| **Learning Outcomes:**  On completing this course successfully you will be able to: A. Code non-trivial programs in an object-oriented programming language. B. Design non-trivial programs using appropriate design methods. C. Apply principles of code design for flexibility and re-use. D. Design and code object-oriented programs. | | | |

|  |
| --- |
| Plagiarism is presenting somebody else's work as your own. It includes: copying information directly from the Web or books without referencing the material; submitting joint coursework as an individual effort; copying another student's coursework; stealing coursework from another student and submitting it as your own work.  Suspected plagiarism will be investigated and if found to have occurred will be dealt with according to the procedures set down by the University. Please see your student handbook for further details of what is / isn't plagiarism. Details are also on the [Student Intranet](https://cms1.gre.ac.uk/student/plagiarism.asp).  **All material copied or amended from any source (e.g. internet, books) must be referenced correctly according to the reference style you are using.   Your work will be submitted for electronic plagiarism checking.  Any attempt to bypass our plagiarism detection systems will be treated as a severe Assessment Offence.** |

#### Coursework Submission Requirements

#### An electronic copy of your work for this coursework must be fully uploaded by midnight on the Deadline Date of Sunday 17/07/2016 using the link on the coursework Teachmat page for COMP1148.

#### For this coursework you must submit a single Acrobat PDF document. In general, any text in the document must not be an image (ie must not be scanned) and would normally be generated from other documents (eg MS Office using "Save As .. PDF"). More details are on the [IT Support pages](http://labs.cms.gre.ac.uk/) .  An exception to this is hand written mathematical notation, but when scanning do ensure the file size is not excessive.

#### For this coursework you must also upload a single ZIP file containing supporting evidence.

#### There are limits on the file size (current values are on TeachMat and the Student Intranet).

#### Make sure that any files you upload are virus-free and not protected by a password or corrupted otherwise they will be treated as null submissions.

#### Your work will be marked online and comments on your work and a provisional grade will be available from the Coursework page on Teachmat. A news item will be posted when the comments are available, and also when the grade is available in BannerWeb.

#### You must NOT submit a paper copy of this coursework, or include the Banner header sheet.

#### All courseworks must be submitted as above. Under no circumstances can they be accepted by academic staff

**The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences.  See** [**http://www2.gre.ac.uk/current-students/regs**](http://www2.gre.ac.uk/current-students/regs)

### Cinema Ticket Machine Simulation

This is an INDIVIDUAL coursework.

You are asked to produce a simulation of a ticket vending machine that supplies tickets at a cinema. The charges for each ticket are as follows:

VIP seats – £12.90

Front stalls – £9.50

Middle stalls – £7.30

Back stalls – £6.70

However, any system developed should be easily modified to allow for changes in prices charged. The machine can accept the following only – 10p, 20p, 50p, £1.00, £2.00, £5, £10 and £20.

**Stage 1 (max 10% including report):** Design a suitable GUI for this application with sketches of the layout. You may wish to consider examples such as vending machines. You should allow for your design to simulate of input of the payment required including a way of specifying which money is being input. You will also need to include a ‘cancel’ facility that allows the user to reset the machine, returning any payment made.

**Stage 2 (max 20% including report):** Implement the system designed in Stage 1. Having selected a ticket type the user should be prompted for the correct payment; only on completion of payment should the program dispense a ticket. Your program should indicate when the ticket is dispensed; this might be achieved with an appropriate message or a visual indication of the process of dispensing the ticket. You will need to design white box testing of your system and provide evidence of both the functionality of your program and the testing results.

**Stage 3 (max 10% including report):** Modify your implementation, separating the program business function from the GUI using an auxiliary class.

**Stage 4 (max 20% including report):** Implement a change facility which calculates the change due and the money required when the user inserts a payment greater than the charge for the ticket selected. Extend your program further with one of the following features:

* For a transaction the user can buy only one ticket type, but the user can buy more than one ticket, e.g. three front stalls tickets, but NOT a mixture such as two middle stalls tickets and three back stalls tickets.
* Allow the program to store how many tickets of each ticket type are left in the machine so that when it runs out of tickets this can be indicated to the user.

**Stage 5 (max 20% including report):** Extend the program further with an extra auxiliary class TicketLogger.java, which uses a text file or a database to store a record of each transaction including ticket type dispensed, date, time, payment and change dispensed.

**Stage 6 (max 20% including report):** Extend the program to store a count of how many of each note and coin is in the machine. The ticket dispenser should indicate which coins will be used in the change and a ‘no change’ state when it cannot give change. You may use text information or displaying images of coins to show the change dispensed.

### Deliverables

To cover as much of stages 1 – 6 as you have completed:

A zip file containing working code in the form of .java and .class files, or a zipped NetBeans project folder, together with any files or databases you have used for external storage.

A written report, containing the evidence of all completed stages, which should include the following sections:

* Introduction
* A description of how you designed and developed the final code with suitable screen shots of the program in operation
* Details of any faults and failures, including a discussion of the white box testing results.
* Conclusions and reflection. For the reflection you should write at least 400 words, answering the following:

What did I actually achieve with this element of learning? Which were the most difficult parts, and why were they difficult for me? Which were the most straightforward parts, and why did I find these easy?

* Appendices should contain:
  1. Test table and results as detailed in the Testing section of the workbook – this should be updated from the version you submitted for interim deliverable C to cover any changes you have made to the code since then
  2. A full program listing (copy and paste your code into the report as text).

The written part (excluding appendices) should be no more than 2,000 words and there should be no more than 10 screen shots. There will be a penalty for going over either of these limits.

This final deliverable is to be submitted to the coursework upload system BEFORE THE COURSEWORK DEADLINE.

**You *may* also be required to demonstrate your program**.

**Assessment Criteria**

Marks are awarded for:

* The functionality of the programme.
  + Does the system do what it is supposed to do (according to the specification above)?
* Usability:
  + Is your system straightforward and easy to use?
  + Is it obvious to the user what to do?
  + Are all messages clear and unambiguous?
  + Is bad input data handled appropriately?
  + Is the output formatted appropriately?
  + Is the system free from crashes and uncaught exceptions?
* Quality of the Java code:
  + Inclusion of meaningful comments.
  + Use of sensible naming standards.
  + Clear code layout and formatting.
* Quality and completeness of the report:
  + Is the design documentation clear and concise?
  + Have you included evidence of appropriate testing?
  + Have you discussed any faults or failures?
  + Have you reflected on the development process?

**Grading Criteria**

This is a staged assessment. You cannot get marks for a particular stage unless you have made a reasonable attempt at **all** of the previous stages.

|  |  |
| --- | --- |
| % | |
| 86 – 100 | The work examined is exemplary and provides clear evidence of a complete grasp of the knowledge, understanding and skills appropriate to level 4. There is also ample excellent evidence showing that all the learning outcomes of the course are fully satisfied. |
| 76 – 85 | The work examined is outstanding and demonstrates comprehensive knowledge, understanding and skills appropriate to level 4. There is also excellent evidence showing that all the learning outcomes of the course are fully satisfied. |
| 70 – 75 | The work examined is excellent and is evidence of comprehensive knowledge, understanding and skills appropriate to level 4. There is also excellent evidence showing that all the learning outcomes of the course are satisfied. |
| 65 – 69 | The work examined is very good and is evidence of the knowledge, understanding and skills appropriate to level 4. There is also very good evidence showing that all the learning outcomes and responsibilities appropriate to the Level are satisfied. |
| 60 – 64 | The work examined is good and is evidence of the knowledge, understanding and skills appropriate to level 4. There is also good evidence showing that all the learning outcomes of the course are satisfied. |
| 55 – 59 | The work examined is sound and is evidence of the knowledge, understanding and skills appropriate to level 4. There is also sound evidence showing that all the learning outcomes of the course are satisfied. |
| 50 -54 | The work examined is sound but provides limited evidence of the knowledge, understanding and skills appropriate to level 4. There is also sound but limited evidence showing that all the learning outcomes and responsibilities to that Level are satisfied. |
| 45 – 49 | The work examined is acceptable but provides significantly restricted evidence of the knowledge, understanding and skills appropriate to level 4. There is also acceptable but significantly restricted evidence showing that all the learning outcomes of the course are satisfied. |
| 40 – 44 | The work examined is acceptable but provides barely sufficient evidence of the knowledge, understanding and skills appropriate to level 4. There is also acceptable but barely sufficient evidence showing that all the learning outcomes of the course are satisfied. |
| 35 – 39 | The work examined narrowly fails to provide sufficient evidence of the knowledge, understanding and skills appropriate to level 4. There is acceptable evidence showing that the great majority of the learning outcomes of the course are satisfied. |
| 30 – 34 | The work examined provides insufficient evidence of the knowledge, understanding and skills appropriate to level 4. The evidence provided shows that the majority of the learning outcomes of the course are satisfied. |
| 20 – 29 | The work examined is unacceptable and provides little evidence of the knowledge, understanding and skills appropriate to level 4. The evidence shows that only some of the learning outcomes of the course are satisfied. |
| 10 – 19 | The work examined is unacceptable and provides negligible evidence of the knowledge, understanding and skills appropriate to level 4. The evidence shows that few of the learning outcomes of the course are satisfied. |
| 0 – 9 | The work examined is unacceptable and provides no evidence of the knowledge, understanding and skills appropriate to level 4. The evidence fails to show that any of the learning outcomes of the course are satisfied. |