|  |  |  |  |
| --- | --- | --- | --- |
| **COMP1148 (2015/16)** | **Computer Programming** | **Header ID 227775** | **Contribution 50% of course** |
| **Course Leader Dr Chris Walshaw** | **Release Date Monday 09/11/2015** |  | **Deadline Date Sunday 20/03/2016** |
| 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 20/03/2016 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

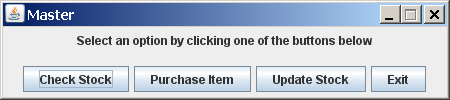
**The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences.  See** [**http://www2.gre.ac.uk/current-students/regs**](http://www2.gre.ac.uk/current-students/regs)

# Stock Control Simulation

You are asked to produce a simulation of a small stock control system for a store. You are provided with the following files as a download:

**Master.java:**

This is what the Master GUI looks like when run:

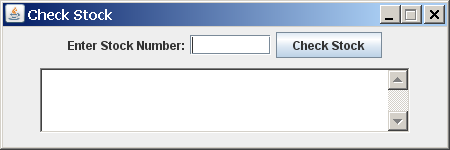


This frame is not resizable and the X button is disabled – the user must click the Exit button to quit.

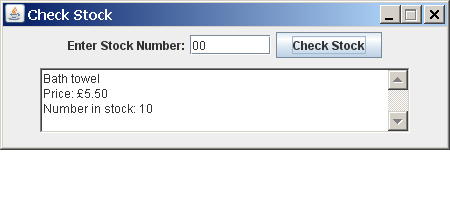
The Purchase Item and Update Stock buttons do nothing at this stage.

**CheckStock.java:**

This is what the Check Stock GUI looks like when run by clicking the Check Stock button on the Master GUI. It is also not resizable but the X button disposes of it. Note that the Master GUI is still displayed on screen:



If the user enters a valid stock number and clicks the Check Stock button, details of the item of stock are displayed:



If the stock number is invalid, an error message is displayed instead.

You will also be supplied with the StockData.java class which holds the stock items as a static data store. You do NOT need to be able to fully understand this class, but note that it provides four public static methods which you may use:

public static String getName(String key) …

public static double getPrice(String key) …

public static int getQuantity(String key) …

public static void update(String key, int extra) …

…

For instance the getName method is used in CheckStock.java to obtain the name of the item with the given key (it returns null if there is no such item in stock).

Using CheckStock.java as an example, you are to write two JFrames, PurchaseItem.java and UpdateStock.java, and complete the functionality for the Master class:

* **PurchaseItem.java** should allow the user to enter a key for an item, and the number of items required. If the key is valid and there are enough items in stock:
  + the user should be presented with a bill for the items
  + the quantity in stock should be updated to reflect the sale

Otherwise a suitable error message should be displayed.

* **UpdateStock.java** should allow the user (in this case, someone working in the warehouse) to enter a key for an item, and the number of new items delivered to the store. If the key is valid a message showing the item name, the number of new items and the number now in store should be displayed, otherwise a suitable error message should be displayed.

You may use the NetBeans GUI builder to help design your JFrames but only as self-directed learning: your tutor will not help you with this.

There are a number of stages to the development and not all students will manage all the deliverables. Each stage will be awarded the marks up to the maximum allocated ONLY when handed in AND accompanied by a relevant section in the final report.

Additional help will be provided in lectures, tutorials and the second workbook for some of the later deliverables.

**Stage 1 Understanding the code (max 10%, inclusive of report)**

**CheckStock.java** should be rewritten with lines of comment explaining exactly how all the code works, e.g.

// import the classes of the abstract windows toolkit and

// swing to enable the use of components such as buttons

import java.awt.\*;

import javax.swing.\*;

**Stage 2 First GUIs designed (max 10%, inclusive of report)**

**PurchaseItem.java** and **UpdateStock.java** should be designed and the code written in prototype form WITHOUT functionality (just the GUI appearance).

**Stage 3 A basic working version (max 20%, inclusive of report)**

Implement the system whose GUIs you designed in stage 2.

**Stage 4 Improving your code (max 20%, inclusive of report)**

Design *white box testing* of your system and provide evidence of both the functionality of your program and the testing results.

Also adapt the code to perform error checking for bad input such as an input quantity of ‘four’.

Adapt the code to include at least two of the following:

* a larger selection of items in **StockData.java**
* displaying pictures of the stock when the user checks it
* a password dialog before the UpdateStock.java JFrame will open
* a redesigned user interface for the JFrames

**Stage 5 Saving data externally (max 20%, inclusive of report)**

Using a database or other storage – more details will be provided later.

**Stage 6 Innovations** (**max 20%, inclusive of report**), for example:

* When updating, include the facility to add new stock lines.
* Use a keypad for input when checking stock availability (see the program **Safe.java** in the second workbook for ideas).
* Multiple purchases (Argos allows up to 5 purchases per transaction). WARNING – this is hard to accomplish if you include full error checking.

**Deliverables**

There are three interim deliverables and a final deliverable which you will need to upload by the given deadlines. All deliverables should be uploaded from Teachmat on the coursework upload system. Failure to demonstrate your interim deliverables is a breach of the coursework regulations and will result in a loss of marks.

**Interim DELIVERABLE A – To Be Uploaded**

**BEFORE MIDNIGHT 6th DECEMBER 2015**

For Deliverable A you should present your commented **CheckStock.java** and the prototype GUIs for **PurchaseItem.java** and **UpdateStock.java** (stages 1 & 2). This should be uploaded before midnight on Sunday 6th December 2015. You will demonstrate your upload and receive verbal feedback from your lab tutor on the following day or the week after.

**Interim DELIVERABLE B – To Be Uploaded**

**BEFORE MIDNIGHT 24th JANUARY 2016**

At this stage you should present a basic working version of the code (stage 3) to be uploaded before midnight on Sunday 24th January 2016. You will demonstrate your upload and receive verbal feedback from your lab tutor on the following day or the week after.

**Interim DELIVERABLE C – To Be Uploaded**

**BEFORE MIDNIGHT 21st February 2016**

You should present a full white box testing plan and code with error checking and some additional features (stage 4). This should be uploaded before midnight Sunday 21st February 2016. You will demonstrate your upload and receive verbal feedback from your lab tutor on the following day or the week after.

**Final Coursework Deliverable – To Be Uploaded**

**BEFORE MIDNIGHT 20th March 2016**

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, answer either (a) or (b) from the following:
  1. 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?
  2. What have I got out of doing this element of the course? How have I developed my knowledge and skills? How do I see this element of the course helping me in the longer term?
* Appendices should contain:
  1. The commented version of **CheckStock.java** that you submitted for interim deliverable A
  2. 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
  3. 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.

**Provided Code** – available to download from Moodle:

* Master.java
* CheckStock.java
* StockData.java

**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?
* Interim deliverables:
  + Have you demonstrated the progression of your work to your lab tutor as required?

**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. |