Informatics and Computer Science	**Coursework Brief Proofing & Printing Confirmation Sheet**
Module Title **Programming in Java**	Module Code **24CSCI04C**
Module Leader **Dr. Khaled Alsheshtawi**	Semester **Two**
Proofed by **Dr. Randa Alanwar**	

I hereby confirm that:

- This coursework brief has been proof-read (spelling and grammar) ☑
- This coursework brief assesses the ILOs for the module ☑
- This coursework brief follows the approved template ☑
- All questions (and sub questions) have their marks specified ☑

Signed (Proof Reader): _______ Dr. Randa Alanwar ___________

Signed (Module Leader)______ Dr. Khaled Alsheshtawi ___________

Informatics and Computer Science	**24CSCI04C** **Assignment 2** **2024/2025**
Module Title **Programming in Java**	
Module Leader **Dr. Khaled Alsheshtawi**	Semester **Two**
Assessment Weight **25 % of the total course mark**	Due Date **Week 10**

Instructions to students:

1. This is a group assignment; each group consists of (4 to 6) students. Kindly follow the instructions of the TAs regarding registration of the group.

2. Submission: The submission is via the e-learning system only, by 12:00 pm on the deadline day.

3. Assessment: Assessment evaluates the class diagram, the submitted java program code, oral discussions, and teamwork performance.

4. Feedback: Will be given through generic feedback in labs and specific written feedback for each group two weeks after submission.

5. Along with the submitted project, you need to submit: a fully completed and signed *Coursework submission form and* a *Statement of Academic Honesty Form*. You can only submit your own work. Any student suspected of plagiarism will be subject to the procedures set out in the GAR.

6. AI tools and online resources are allowed; however, their use must not exceed 30%. Students must specify the tools and resources used, the reasons for their usage, and the specific benefits gained from them.

I. Assignment Objective:

Finalize your selected system by implementing the functionalities that make the application user friendly and functional. Each Team member should choose at least one user type that has access to the system. At the start of the application the user needs to create an account or login if they already have an account. According to the log in category, the interface is selected such that only the functionalities allowed for this user type are displayed.

Requirements: GUI, Files and Exception Handling

1) Implement the GUI for the user you represent to enable access to the functionalities of that user (functionalities from assignment 1 and any other extra functionalities you deem necessary).

2) Implement saving/loading your data to/from files [Sample files include : Log-in Info, ….]. Choose from these files according to your user type and add any other files you need.

3) Implement error-handling mechanism through Java Exception handling. Each Team member should implement a user defined exception handling.

II. Submission: Via the e-Learning system, each group needs to submit a pdf file and their source code as follows:

1. Team members and a description of work division among them.

2. Detailed class diagram.

3. A zipped project file with the Java source code.

III. Deadlines and Deliverables:

You need to decide on the working order of your chosen system, and hence select the classes you will utilize. On Week 10, you are to submit the group members and your selected project.

IV. Grading Standards:

Every criterion will make up an approximate percentage of the grade given to a single programming problem as indicated in the "Approx. % of Grade". Points will be assigned for a particular criterion roughly along the lines of the guidelines of the "Excellent," "Good," "Satisfactory," and "Not Met" evaluations.

For example, a team assignment that was marked as "Good" in the Design Enhancements, "Satisfactory" for GUI criterion, and "Excellent" in all other areas would receive: 0.8*0.1 + 0.8*0.1 + 1*0.2 + 0.8*0.2 + 1*0.1 + 1*0.1 = 88%.

****** As a special case, if a program does not meet the specifications at all / is entirely incorrect, no credit will be received for the other criteria either.***

1. **<u>Design Enhancements</u>: (Approx. % of Grade: 10%)**

   1.1. (**Excellent**: **100**%):
   - Completeness between 90% and 100% of the design enhancements.

   1.2. (**Good**: **80**% of the points):
   - Completeness between 80% and 90% of the design enhancements.

   1.3. (**Satisfactory**: **60**%):
   - Completeness between 70% and 80% of the design enhancements.

   1.4. (**Not Met**: **<40**%):
   - Completeness less than 70% of the design enhancements.

2. **<u>Authentication and Authorisation</u>: (Approx. % of Grade: 10%)**

   2.1. (**Excellent**: **100**%):
   - Satisfies the user name, password, Limits failed attempts to three before notifying the user and exiting the program, Gives authenticated users access to the correct role file after successful authentication, Allows a user to log out, Stays on the credential screen until either a successful attempt has been made, three unsuccessful attempts have been made, or a user chooses to exit.

   2.2. (**Good**: **80**%):
   - Satisfies most of the: Gives authenticated users access to the correct role file after successful authentication, Allows a user to log out, Stays on the credential screen until either a successful attempt has been made, three unsuccessful attempts have been made, or a user chooses to exit

   2.3. (**Satisfactory**: **60**%):
   - Suffers from only one of the following: Gives authenticated users access to the correct role file after successful authentication, Allows a user to log out, Stays on the credential screen until either a successful attempt has been made, three unsuccessful attempts have been made, or a user chooses to exit

   2.4. (**Not Met**: **<40**%):
   - Suffers from more than one of the following: Gives authenticated users access to the correct role file after successful authentication, Allows a user to log out, Stays on the credential screen until either a successful attempt has been made, three unsuccessful attempts have been made, or a user chooses to exit

3. **<u>GUI</u>: (Approx. % of Grade: 10%)**

   3.1. (**Excellent**: **100**%):
   - GUI satisfies all of the following: visually attractive objects, compatible colours, consistent software components, great alignment, sizing and spacing, standard software interface conventions followed

   3.2. (**Good**: **80**%):
   - GUI satisfies most of the following: visually attractive objects, compatible colours, consistent software components, great alignment, sizing and spacing, standard software interface conventions followed

   3.3. (**Satisfactory**: **60**%):
   - GUI suffers from only one of the following: visually distracting objects, clashing colours, inconsistent software components, poor alignment, sizing or spacing, non-standard software interface conventions.

3.4. (**Not Met**: **<40**%):

- GUI suffers from more than one of the following: visually distracting objects, clashing colours, inconsistent graphical components, poor alignment, sizing or spacing, non-standard software interface conventions.

4. Files & I/O: (Approx. % of Grade: 20%)

4.1. (**Excellent**: **100**%):

- File is correctly formatted and read by the driver.

4.2. (**Good**: **80**%):

- N/A

4.3. (**Satisfactory**: **60**%):

- N/A

4.4. (**Not Met**: **<40**%):

File(s) not used properly.

5. REST Web Services with JAX-RS: (Approx. % of Grade: 20%)

5.1. (**Excellent**: **100**%):

- File is correctly formatted and read by the driver.

5.2. (**Good**: **80**%):

- N/A

5.3. (**Satisfactory**: **60**%):

- N/A

5.4. (**Not Met**: **<40**%):

File(s) not used properly.

6. Exception-Handling: (Approx. % of Grade: 20%)

6.1. (**Excellent**: **100**%):

- Exception handling is implemented correctly and appropriately using try/catch blocks in order to create a more robust program. At least one exception handled.

6.2. (**Good**: **80**%):

- Use most of the necessary exception handling.

6.3. (**Satisfactory**: **60**%):

- Suffers from the necessary exception-handling

6.4. (**Not Met**: **<40**%):

- Suffers from not using the exception-handling

7. Oral Presentation / Fielding Questions: (Approx. % of Grade: 10%)

7.1. (**Excellent**: **100**%):

- The team present their program to the TAs clearly, confidently, and accurately on all portions of the code and its operation.
- The team answer all questions from the TAs about the program and its operation clearly, smoothly, confidently, and convincingly.
- The team provide alternative solutions.

7.2. (**Good**: **80**%):

- The team present their program to the TAs in an understandable way, with slight hesitations, and mostly accurate on all portions of the code and its operation.
- The team answer all questions from the TAs about the program and its operation clearly, smoothly, confidently, and convincingly.
- The team explain the entire program correctly as it is.

7.3. (**Satisfactory**: **60**%):

- The team present their program to the TAs in an slightly confused way, with noticeable hesitations, and occasionally accurate on portions of the code and its operation.
- The team answer all questions from the TAs about the program and its operation somewhat clearly, slight hesitations, confident but reserved, and most times convincingly.
- The team explain a little program design.

7.4. (**Not Met**: **<40**%):

- Student presentation of their program to the TA is not comprehensible, many hesitations, and not accurate with any portion of the code and its operation.
- The team answer only a few questions from the TA about the program and its operation and the answers are lacking in clarity, there are long hesitations, they are unconvincing.
- The team is unable to explain the program design.

Module Specification – (Programme Specs Ver. 6.0)

Module Code: CSCI04C	**Title:** Programming in Java	
Level: C	**Modular weight:** 10	**Faculty/Dept.:** ICS/CS
Pre-requisite modules: CSCI02P – Introduction to Programming and Problem Solving		
Reassessment: No restriction		
Module Leader: Dr. Khaled Alsheshtawi		
Semester taught: Two		
Date of latest revision: March 2022		

Aims

The aim of this module is to provide basic grounding in Java language programming and lay a firm foundation from which other modules, that require Java programming, may rely upon. Students will learn to write moderately complex Java code, understand the basics of the language API, and development practices (the object-oriented development process, test-driven development, and refactoring).

Intended Learning Outcomes

On completion of this module students should be able to:

Knowledge and understanding

1. Define concepts, principles and philosophy of Java language, and the Java technologies relating to web applications. [A2]

Intellectual Skills

2. Develop an analytical approach to interpreting intermediate problem specifications and a systematic approach to problem solving. [B2]

Practical and Professional Skills

3. Design and develop solutions to a number of intermediate programming problems using Java. [C2]
4. Use Java API (AWT, Swing, Collection…etc.) efficiently to solve intermediate programming problems. [C7]

General and Transferable Skills

5. Develop interpersonal skill of self-learning, planning, time management, and communication on individual and team programming projects. [D1, D2]

Employability

This module will provide opportunities for students to:

1. Understand the importance of being self-motivated in order to progress the area of work. [A5].
2. Carry out a range of complex ICT activities related to their work that involve application software. [B.3.1]
3. demonstrate effective planning, prioritisation and organisation to plan activities and carry them through effectively [C.2.2]

4. demonstrate effective team working by building and developing appropriate relationships with academic staff, peers, colleagues and people within the organisation. [C.3.2]

Indicative Content

- Java classes and methods.
- OOP concepts in Java.
- Graphical User Interface Interfaces. I/O in Java.
- Collection framework.
- Handling events and exceptions.
- Java applications & applets.
- Unit Testing

Methods of Learning, Teaching and Assessment

Total student effort for the module: 100 hours on average over one semester.

Type of session	ILOs Covered	Typical Student Effort		
		Typical number in the semester/s	Typical hours per week	Total hours
Lecture	1-4	12	2	24
Tutorial	-	-	-	-
Laboratory	2-5	12	2	24
Private study	1-4	-	-	52

Assessment

Assessment Type	Weight %	ILOs Assessed	Exam Semester	Exam/ Written Coursework Length
Two group programming assignments, (4 - 6 students).	50	3-5	2	-
One unseen written exam	50	1-3	2	2 hours

Methods of Feedback

In response to assessed work:

- Specific written feedback will be provided for assignment one in written form and returned to students along the marked coursework.
- Generic feedback in laboratory for assignment two and marked coursework with the marking schema published on E-Learning.
- Generic feedback for unseen exam on e-learning.

Developmental feedback generated through teaching activities:

- Dialogue between students and staff in labs and lectures

Indicative Reading List

- Paul Deitel and Harvey Deitel. Java How to Program, 11th edition. Prentice Hall, 2022.
- Liang, Y Daniel. Introduction to Java Programming-Comprehensive Version, 12th Edition. Pearson, 2019.