جامعة الحسين التقنية
Al Hussein Technical University

ASSIGNMENT BRIEF

HTU Course No: 40201100	**HTU Course Name:** Programming
BTEC Unit Code: H/618/7388	**BTEC UNIT Name:** Programming

Version: 3

Student Name/ID Number/Section	
HTU Course Number and Title	40201100 Programming
BTEC Unit Code and Title	H/618/7388 Programming
Academic Year	2023-2024 Spring
Assignment Author	Dania Alsa`id
Course Tutor	Dania Alsa`id - Sultan Alrushdan - Hana' Rashid - Abeer Malkawi - Mariam Biltawi - Rami Al-Ouran - Shatha Hawawsha
Assignment Title	Programming using Java
Assignment Ref No	1
Issue Date	04/05/2024
Formative Assessment dates	From 05/05/2024 to 30/05/2024
Submission Date	08/06/2024
IV Name & Date	Malek Louzi 03/05/2024

Submission Format

The submission for this assignment is divided into:

- An individual written report (see report guidelines below) that includes every item outlined in the assignment details supported with screenshots.
- A full working software program. The uploaded source code (.java files) should include every item in the assignment details that requires code writing.
- Declaration form filled and signed.
- An oral discussion to discuss the assignment submission.

Report guidelines:

- The report should be submitted as DOCx softcopy to the university's eLearning system within the deadline specified above.
- You should name your report as follows: Yourname_YourNumber_Programming.
- Your report should be written in a formal business style using single spacing and font size 12 of times roman.
- In your report, you should make use of headings, paragraphs, and subsections as appropriate.
- Your report must be supported with research and referenced using the Harvard referencing system.
- The citation percentage should not exceed 20%; otherwise, you will be penalized. Plagiarism is forbidden.

General Rules:

- Late submission will not be accepted.
- Submission via email or teams are not allowed.
- Oral discussion plays a crucial role in forming the student's mark.

Unit Learning Outcomes

LO1 Define basic algorithms to carry out an operation and outline the process of programming an application

LO2 Explain the characteristics of procedural, object-orientated and event-driven programming

LO3 Implement basic algorithms in code using an IDE

LO4 Determine the debugging process and explain the importance of a coding standard

Assignment Brief and Guidance

Scenario:

You are applying for an internship opportunity at a well-known company. As part of the application process, they have requested that you submit a task to determine your eligibility for an interview.

The task is as follows:

Part 1:

1. Describe the process of building an application.
2. What is your understanding of an algorithm, explain the importance of algorithms in computer programming.
3. How do you ensure that an algorithm is efficient and effective.
4. Outline the steps involved in writing code for a program.
5. Describe the process of compiling code into machine-readable instructions.
6. Provide an example of an algorithm and clearly explain it. Then, implement the algorithm using a programming language. You will include this algorithm in the paradigms below.

Part 2:

1. Define the procedural programming paradigm and its key characteristics.
2. Explain the concept of object-oriented programming (OOP) paradigm and its fundamental principles.
3. Describe the event-driven programming paradigm and how it differs from procedural and object-oriented paradigms.
4. Provide examples of code that primarily demonstrate the three paradigms mentioned above, showcasing their core features.
5. Compare and contrast procedural, object-oriented, and event-driven paradigms in terms of their approach to programming (Support your claims using screenshots of your code):
    - Describe how the source codes utilize each paradigm.
    - Reflect on how the chosen programming paradigms align with the overall goals and requirements of your application.
6. Provide a complete scenario where a combination of these paradigms might be beneficial in software development. Include the previously written algorithm in your program.
7. How do you ensure that your code is well-documented and readable by others?
8. Explain the importance of testing code before execution.

Part3:

1. Describe the steps you take to create a new project in your preferred IDE. How do you organize project files and directories within the IDE?
2. What features or tools provided by IDEs do you find most beneficial for your development workflow?
3. Describe any specific IDE debugging features you find helpful.
4. Compare the overall development experience, including speed, efficiency, and ease of use, between using an IDE and not using one.

Part4:

1. Discuss the error types and how each error affects the overall application.
2. Walk through the general steps involved in the debugging process when you encounter a bug in your code.
3. Explain the role of breakpoints in debugging and how they help in identifying and resolving issues in code.
4. How does the debugging process contribute to the development of secure applications? Can you provide examples of security vulnerabilities that debugging helps identify and address?
5. Reflect on any experiences where the debugging process played a crucial role in uncovering security weaknesses or vulnerabilities in your applications. How did you address these issues?

Part5:

1. What coding standards do you follow when writing code.
2. Describe any specific naming conventions prescribed by the coding standard you follow. How do these conventions help improve code readability and maintainability.
3. Why do you believe having a coding standard is important for a development team? What benefits does it offer in terms of collaboration, code quality, and project success.

Tips and Tricks for the Oral discussion questions

1. Describe the overall structure of the source code. How is it organized, and what are the main components or modules?
2. Identify specific sections of the code that adhere to the procedural programming paradigm. How are procedures or functions used, and what is their role in the application?
3. Analyze the object-oriented aspects of the code. Identify classes, objects, and inheritance relationships. How are encapsulation, inheritance, and polymorphism utilized?
4. Evaluate the clarity and readability of the code. Are variable names, comments, and documentation clear and meaningful? How does this impact understanding?
5. Assess the modularity and reusability of the codebase. Are there separate modules or components that can be easily reused or modified without affecting other parts of the application?
6. Consider the event-driven aspects of the code. How are events triggered, handled, and propagated throughout the application? Are event handlers implemented effectively?
7. Discuss the maintainability of the codebase. Is it easy to make changes or additions to the code without introducing errors or unintended side effects?

8. Identify any potential security vulnerabilities in the code. Are input validation, error handling, and data protection mechanisms implemented effectively?
9. Reflect on the overall quality of the code. Does it adhere to best practices and coding standards? Are there areas for improvement?
10. Consider the alignment of the codebase with the requirements and goals of the application. Does the code effectively meet the intended functionality and user needs?

Learning Outcomes and Assessment Criteria			
Learning Outcome	**Pass**	**Merit**	**Distinction**
LO1 Define basic algorithms to carry out an operation and outline the process of programming an application	**P1** Define an algorithm and outline the process in building an application. **P2** Determine the steps taken from writing code to execution.	**M1** Analyse the process of writing code, including the potential challenges faced.	**D1** Evaluate the implementation of an algorithm in a suitable language and the relationship between the written algorithm and the code variant.
LO2 Explain the characteristics of procedural, object-orientated and event-driven programming	**P3** Discuss what procedural, object oriented and event driven paradigms are; their characteristics and the relationship between them.	**M2** Compare the procedural, object-orientated and event driven paradigms used in given source code of an application.	**D2** Critically evaluate the source code of an application that implements the procedural, object-orientated and event-driven paradigms, in terms of the code structure and characteristics.
LO3 Implement basic algorithms in code using an IDE	**P4** Write a program that implements an algorithm using an IDE.	**M3** Enhance the algorithm written, using the features of the IDE to manage the development process.	**D3** Evaluate the use of an IDE for development of applications contrasted with not using an IDE.
LO4 Determine the debugging process and explain the importance of a coding standard	**P5** Explain the debugging process and the debugging facilities available in the IDE. **P6** Explain the coding standard you have used in your code.	**M4** Examine how the debugging process can be used to help develop more secure, robust applications.	**D4** Evaluate the role and purpose of a coding standard and why it is necessary in a team as well as for the individual.