| **Student number:** **22203957** | **https://github.com/sbonelogit/tp2\_tesk1** | **Total mark (%):** | | **46/120=38%** |
| --- | --- | --- | --- | --- |
| **Criteria/Task** | **Description/Expectations** | **Points** | **mark** | **Comment (if applicable)** |
| **Overall Requirements and Preparation** | **Preparation includes installed software, accounts setup, and compliance with test requirements.** | **5** | **5** |  |
| **Question 1: Git and GitHub for Version Control** |  | **22** |  |  |
| 1.1 Create a directory "tp2\_test\_1" | Directory creation and organization | 3 | 2 |  |
| 1.2 Create "readMe.md" with SCM lifecycle explanation | Explanation clarity, use of git commands, and example relevance | 8 | 2 |  |
| 1.3 Initialize a git repository locally | Correct initialization | 2 | 2 |  |
| 1.4 Create a branch and merge changes | Branch creation, file addition, and successful merge | 7 | 0 |  |
| 1.5 Create a cloud repository and add collaborator | Repository setup and collaborator addition | 2 | 2 |  |
| **SUBTOTAL** |  |  |  |  |
| **Question 2: Data Structures and Algorithms** |  | **61** |  |  |
| 2.1 Write "register\_party" function | Logic, adherence to requirements, and coding best practices | 15 | 13 |  |
| 2.2 Register MK party | Correct application of function and adherence to IEC regulations | 6 | 4 |  |
| 2.3 Implement "update\_voter\_info" function | Function implementation, coding best practices, and git usage | 13 | 0 |  |
| **SUBTOTAL** |  |  |  |  |
| 2.4 Filter parties using list comprehension and filter function | Correct use of list comprehension and filter | 10 | 0 |  |
| 2.5 Rewrite list comprehension into normal list | Correct translation and list creation | 8 | 0 |  |
| 2.6 Extract registered voters using lambda function | Correct use of lambda and filter functions | 6 | 0 |  |
| 2.7 Update "readMe.md" with sets data structure explanation | Clarity of explanation | 3 | 0 |  |
| **SUBTOTAL** |  |  |  |  |
| **Question 3: Basic on Data Insights (Programming Perspective)** |  | **24** |  |  |
| 3.1 Load "football.csv" into a dataframe | Correct data loading | 5 | **0** |  |
| 3.2 Inspect dataframe (last 7 rows) | Correct inspection and display | 2 | **0** |  |
| 3.3 Various data access and manipulation tasks | Correctness and clarity in each sub-task | 14 |  |  |
| 3.3.1 | Display the **"Nationality"** and **"Club"** columns for the all players | 2 | **0** |  |
| 3.3.2 | display the data for the tenth player | 2 | **0** |  |
| 3.3.3 | display the "**Goals"** and **"Appearances**" for index 100 to 110 | 2 | **0** |  |
| 3.3.4 | Add a new column named "**Goals per Appearance"** | 3 | **0** |  |
| 3.3.5 | display a subset of the data frame containing only the players from **"Arsenal"** club | 2 | **0** |  |
| 3.3.6 | filtering operation to display players who have scored more than 5 | 3 | **0** |  |
| 3.4 Upgrade pandas module | Correct upgrade command | 2 | **0** |  |
|  |  |  |  |  |
| 3.5 Push project to cloud repository and confirm collaborator | Successful push and collaborator confirmation | 1 | **1** |  |
| **Documentation and Code Quality** | Clarity, organization, and commenting of code; adherence to software development best practices | 10 | **10** |  |
| **Commit Regularity and Screenshots,** | Frequency and appropriateness of commits; inclusion of required screenshots | **5** | **5** |  |
| **Total** |  | **120** |  |  |