

|  |  |
| --- | --- |
| Module name and code | Computer Science Fundamentals  4BUIS012C-n |
| CW number and weighting | CW 1, 40% |
| Lecturer setting the task with contact details and office hours | Olga Yugay, [oyugay@wiut.uz](mailto:oyugay@wiut.uz)  Discussion group for Q&A,  <https://intranet.wiut.uz/LearningMaterial/Discussion/Details/2635?moduleId=559>  Office hours: Thursday 11:00 - 12:00 |
| Submission deadline | 16 November 2023 |
| Results date and type of feedback | 22 December 2024  Electronic feedback |
| **The CW checks the following learning outcomes:** | |
| 1. Identify and interpret the numerical processes running within the computer system.  2. Describe the tasks and features of each of the main components of a computer system and how these components work together.  4. Explore and experiment with algorithms and other relevant computer science topics.  5. Explore current trends in technology and learn to use relevant resources and tools  6. Demonstrate a thorough knowledge of basic command line commands and git | |

**Task**

General guidelines:

* The coursework must be submitted online through an intranet system.
* The coursework deadline is not negotiable. Cheating and plagiarism carry severe penalties.
* Your work must be properly referenced. All sources - including material from the web – must be acknowledged.

**Task overview**

The coursework requires you to produce a written report on the given task using GoogleDocs and publish relevant files to the GitHub repository. You should complete the assignment **individually or in pairs**. If the work is done in pairs there has to be clear evidence of contribution from both students in both Google doc (fairly equal contributions from students) and GitHub repository (more than 3 commits per student).

|  |  |
| --- | --- |
| Deadline for choosing pair and starting the work on CW to show progress | 17 October 2023 |
| Link for filling out details: | <https://docs.google.com/forms/d/e/1FAIpQLSc0HJCEuwfQLQWAJtCnlcL0wxFjVMVwxLHMRWjmgcDUiCdpzg/viewform?usp=pp_url> |

Recommended word limit: **1000** words excluding information in tables, references.

To produce coursework:

**Part A:**

1. Analyse the given case.
2. Demonstrate your IT knowledge and programming skills using software solutions.
3. Write a report summarising the findings using Google Docs demonstrating the progress. Google **account name should contain your ID number**. Share with [wiut.tutor@gmail.com](mailto:wiut.tutor@gmail.com) in an editor role.

**Part B:**

1. Create GitHub account **(choose a free option)** with a **username that contains your ID number**
2. Access <https://classroom.github.com/a/3hCjVm-L> and accept the assignment. Follow the instructions listed on the page. Create a team named with IDs of team members. Once the assignment is accepted, refresh the page, and access the created GitHub repository link.
3. Once the repository link is accessed follow instructions to set up a new repository or push an existing repository from the command line.
4. Work on the coding part of coursework and push updates to the GitHub repository before the submission deadline. The code files should also be submitted along with the coursework file in a zipped file through the WIUT intranet system.

**Case scenario**

Review a case: You need to demonstrate your programming knowledge and other relevant IT knowledge and skills for a development company. A development startup company has the following entry task that needs to be accomplished within a set deadline.

The task consists of several parts:

1. Dataset Selection:
   * Each team (two people maximum) or a student has to select a unique dataset related to education statistics from Kaggle (<https://www.kaggle.com/datasets>) or similar resources. To make sure the dataset is not taken enter yours in [this form](https://forms.gle/nooTHmnDxpeYEmLy5) and check that it does not repeat in the [following list](https://docs.google.com/spreadsheets/d/e/2PACX-1vRjyI0UZSNF2NG18RCBbsSSHunt2QFehf1XHuDU6flUjtoSIp-p88oH9Td168PUUBbb4dsKHw0ZEKgJ/pubhtml). The first team or student reserves a right over the dataset specified. The deadline for specifying dataset is the same as for specifying the team or individual work.
   * Ensure that each student or team should select a unique dataset to avoid close collaboration. **20 marks will be deducted** for using the same dataset.
2. Project Management:
   * plan a development plan for a chosen project on a cloud-based tool like Trello or Jira. The project has to have a meaningful name itself and for the tasks and deadlines set at least ***once a week starting from 17 Oct.***
   * The project should follow a particular project development lifecycle and have a certain number of tasks with the assigned executor. Set a [wiut.tutor@gmail.com](mailto:wiut.tutor@gmail.com) to be a watcher
3. Code Implementation:
   * You are required to write an individual Python code that performs the following tasks using their selected dataset:

a) Reading the Dataset:

* + Use appropriate modules to read the dataset from the selected CSV file.
  + Store the dataset in a suitable data structure (e.g., list, dictionary, or pandas DataFrame).

b) Data Exploration:

* + Provide a brief description of the dataset, including the number of rows, columns, and available features.
  + Display a sample of the dataset to showcase the structure of the data.

c) Data Cleaning:

* + Identify and handle missing values, if any, by either removing the rows or filling in the missing values with appropriate strategies. Write functions for higher marks.
  + Perform any necessary data transformations, e.g., converting data types.

d) Data Analysis and functions:

* + Perform at least 5 different data analyses on selected datasets using conditionals, loops, individually written functions and libraries.
  + Examples of analyses can include:
    - 1. Calculating the average of specific columns.
      2. Finding the minimum and maximum values of a column.
      3. Counting the occurrences of a specific value in a column.
      4. Grouping the data by a specific category and calculating summary statistics.
      5. Plotting graphs or visualisations to represent the data.
      6. Implement error handling to handle potential issues, such as missing or invalid data.
      7. Implement a user interface that allows the user to interact with the program and perform various analyses.

e) Interface to interact with data you are free to use any of the options below:

* + Command Line Interface (CLI): A CLI interface allows users to interact with a program through a command line interface. Users can provide input and receive output by typing commands and arguments in the terminal. Python provides built-in modules like argparse that can be used to create command line interfaces for data processing programs.
  + Graphical User Interface (GUI): A GUI interface provides a visual interface for users to interact with a program. GUIs are typically used for programs that require more complex user interactions or visualizations. Python provides several libraries for creating GUI interfaces, such as Tkinter, PyQt, and PySide.

f) Code Documentation:

* + Provide proper comments and documentation throughout the code, explaining the purpose and functionality of each section.
  + Follow appropriate naming conventions for variables, functions, and modules.

1. Report
   * A report to be produced should include:
     + Introduction (describe the case, project name, dataset details, any other assumptions)
     + Computer specifications you would require having at your position at starting position with justification.
     + Software you would need to be installed with related version and justification. Specify SaaS/PaaS/IaaS model(s) of the chosen cloud-based solutions where applicable.
     + Project development lifecycle (describe chosen lifecycle and justify its choice)
     + Conclusion (make conclusions based on the recommendations given above; suggest further actions and arrangements)

***Expected Output***: The program should provide clear and concise output for each analysis or function. The user interface should allow the user to input their choices and display the results accordingly. The report needs to be properly formatted and well organised. Justification part should include advantages and disadvantages and suggest alternatives.

**Part A comments: *Google docs instructions***

Work on the coursework under a Google account named with your ID. For example, [0000XXXX@gmail.com](mailto:0000XXXX@gmail.com), where XXXX is your ID. Add [wiut.tutor@gmail.com](mailto:wiut.tutor@gmail.com) to Google Document version of the report as collaborator in editor mode. For submission the coursework document should be downloaded as .docx file and submitted through intranet.

There should be clear evidence of:

* work started early (e.g. at least before 17 October)
* progress made (named version history)
* detailed history of edits
* use of comments

**Part B comments: *Git instructions***

Demonstrate how you can use git to track changes made to code files. Keep working under a repository automatically generated when you accept an assignment by following the link <https://classroom.github.com/a/3hCjVm-L> . You will be keeping track of changes made to files while working on code in part B of the coursework. The name of the repository should contain your ID number(s). Your username should be your ID. If you want to use your existing account, it should not contain any reference to your real name but have an ID in name or description. Make sure your GitHub repository is **private.** The project should have evidence of several (at least 3) commits. Demonstrate use of commit, branch, merge. Push the updates to GitHub and specify a link to your repository in the report. Follow the naming conventions for commits, more about it <https://www.conventionalcommits.org/en/v1.0.0/>.

Check that the CW has a standard cover page, table of contents, page numbers and bibliography. Your name should not appear on the cover page or anywhere else. Put your ID number on the cover page and on every other page.

**Useful sources (consider using other resources as well)**

https://www.wired.com/story/how-to-buy-the-right-laptop-for-you/

https://archa.uz/en/

<https://mediapark.uz/>

<https://www.g2.com/>

https://alternativeto.net/

https://www.capterra.com/

**Recommended structure of the report:**

1. Introduction
2. Review of hardware/software/tools
3. Project Development Lifecycle
4. Justification and recommendations
5. Conclusion
6. Links: Google doc, GitHub
7. References

**Format**

1. Word-processed Times New Roman/ Arial 12, double-spaced.

2. The cover sheet should state your ID number, module title and marker’s name, available at <http://sp.wiut.uz/Shared%20Documents/Forms/AllItems.aspx>

3. Recommended word count: 1500.

4. Include a contents page giving the headings and page numbers of each section.

5. Pages should be numbered.

6. Word count must be included on the cover page (do not include attachments and bibliography in the word count).

7. Please do not submit any loose pages.

8. Use [numbered referencing style](https://www.citethisforme.com/cell-numeric/source-type)

**Submission requirements**

Electronic submission only. No hard copy is required.

Coursework must be submitted electronically through the intranet. Log in under your account, find the Computer science fundamentals module and find designated places on intranet for submission.

Put your report and .py file into the .zip archive folder. Archive should be submitted electronically to the corresponding place in the intranet.

**In addition, the report and .py file should be submitted to a designated place on the intranet to be checked for plagiarism.**

**File naming conventions**

Always save the file as follows: **Modulename.NumberofCW.IDnumber**

Example: **CSF.CW1.467**

This is your responsibility to put CW through the anti-plagiarism software before submission.

**Assessment criteria/marking scheme**

The report will be marked according to the following criteria:

* The quality of the submitted report.
* There is good evidence of the research performed in order to complete the task.
* There is evidence of critical thinking demonstrated in evaluation (justification) of the recommendations

All referenced sources should be acknowledged.

|  |  |
| --- | --- |
|  | **Weight (marks)** |
| **PART 1 Report** | **45** |
| **1. Case introduction** | **5** |
| Describe the case and specific needs assumed by the chosen team. Assumptions should be fully outlined to get a full mark. Assumptions should be fairly well described to get a passing mark.  Description of the case, project name, dataset details, any other assumptions should be present | 5 |
| **2. Review of hardware/software/tools** | **10** |
| *Relevant hardware specifications should be listed.* | 5 |
| *Review of software*  Description of solutions. Brief and to the point description of 3 or more relevant solutions per development process category should be provided to get a fullmark. Fair description of at least 2 relevant solutions per category should be given to get a passing mark for the section. Benefits and limitations are carefully considered. | 5 |
| **3. Project Development Lifecycle**  The development lifecycle is stated, described, and justified in the report | 5 |
| The development lifecycle steps are reflected on Trello website properly. If work in in pair is in involved, it should be reflected. | 5 |
| **4. Justification and recommendation** To get full mark: full justification is done taking into consideration the specifics of a given case and relevant recommendations are given. To get a passing mark for this section: fairly justified taking into consideration only some of the specifics. | 5 |
| **5. Conclusion** To get full mark: relevant conclusions are made taking into consideration the specifics of a given case and relevant recommendations are given. To get a passing mark for this section: fair attempt to make conclusions taking into consideration only some of the specifics of the case/solutions | **5** |
| **6. Google Docs:** To get full mark: detailed planning started early (e.g. at least before 17 Oct): version history, detailed history of edits, comments. To get a passing mark for this section: fair attempt to plan and keep track of changes is present. | **5** |
| **7. General requirements.** General requirements are fully satisfied to get a full mark. General requirements are fairly well satisfied to get a passing mark. | **5** |
| 7.1 Presentation of the report. Report should be well formatted to get a full mark. Report should be fairly well presented to get a passing mark. | 2 |
| 7.2 Referencing. Good range of current references should be given to get a full mark. Fair range of fairly current references should be provided to get a passing mark. | 3 |
| **PART 2** | **50** |
| **Code Implementation:** To get a full mark for the code part the solutions provided should be correct and well commented. Moreover, code shows the correct use of proper variables, data types (list, tuple, dictionary or set), conditionals, loops, functions, modules. Dataset is appropriate.  To get a passing mark for this section: fair attempt to produce the solutions is evident. There is a fair attempt to use at least 3 different data types, conditionals, loop, function. There is evidence to code contributions from both students from git commits. In case of no evidence marks will be decreased.  **20 marks deducted if the same dataset is used.**  **The student will be called for Viva if suspected of plagiarism.** | **30** |
| **Reading the Dataset**: Use appropriate modules to read the dataset from the selected CSV file. Store the dataset in a suitable data structure (e.g., list, dictionary, or pandas DataFrame and others). | 5 |
| **Data Exploration**: Provide a brief description of the dataset, including the number of rows, columns, and available features. Display a sample of the dataset to showcase the structure of the data. | 5 |
| **Data Cleaning**: Identify and handle missing values, if any, by either removing the rows or filling in the missing values with appropriate strategies. Write functions for higher marks. Perform any necessary data transformations, e.g. converting data types. | 5 |
| **Data Analysis and functions:** at least 4 different data analyses on selected datasets using conditionals, loops, individually written functions and libraries correctly for pass mark | 5 |
| **Interface:** cmd interface using argparse or GUI Tkinter, PyQt, and PySide | 5 |
| **Code Documentation:** Provide proper comments and documentation throughout the code, explaining the purpose and functionality of each section. Follow appropriate naming conventions for variables, functions, and modules. | 5 |
| **Git:** To get a **full mark** there should be evidence of correct use of git commands: multiple commits, branch, merge.  To get a **passing mark for this section**: attempt to produce at least **3** commits is present per student. | **20** |

|  |  |  |
| --- | --- | --- |
| **Criteria**  **Mark (%)** | **Part A** | **Part B** |
| **90 – 100** | Professional delivery of introduction and assumed needs. Excellent justification. Brilliant review of software and hardware solutions and tools taking into consideration various range from referenced sources. Full justification and detailed relevant recommendations are given at a professional level. Conclusion is made summarising all the main points taking into consideration the specifics of a given case and relevant recommendations are given. Can be published as part of an academic article or business paper or recommended as a professional report | Professional and realistic development  Code part is correct and properly commented. Code shows the correct use of proper variables, data types (list, tuple, dictionary or set), conditionals, loops, functions, modules. Code executes without errors. Functional or OOP is properly implemented. |
| **80 – 89** | Well focused introduction and assumed needs. Strong justification. Excellent review of relevant software and hardware solutions and tools takes into consideration various ranges from referenced sources. Clear justification and detailed relevant recommendations are given at professional level. Conclusion is made summarising the main points taking into consideration the specifics of a given case and relevant recommendations are given. With small changes this could be a journal article or recommended as a professional report | Code part is correct and well commented. Code shows the correct use of proper variables, data types (list, tuple, dictionary or set), conditionals, loops, functions, modules. Code executes without errors. Functional or OOP is implemented. With small changes can be recommended as a professional piece of work. |
| **70 – 79** | Introduction and assumed needs are well written. Good justification. Good review of software and hardware solutions and tools taking into consideration various ranges from referenced sources. Broad and in-depth justification and detailed relevant recommendations are given at a professional level. Conclusion is made summarising most of the points taking into consideration the specifics of a given case and relevant recommendations are given. Evidence of professional competence. | The code provided is mostly correct and commented. Code has mostly proper use of variables, data types (list, tuple, dictionary or set), conditionals, loops, function. The code can be further optimised with the use of modules. |
| **60 – 69** | Introduction and assumed needs are clearly stated. Fair justification. Clear review of software and hardware solutions and tools taking into consideration various ranges from referenced sources. Fair justification and detailed relevant recommendations are given. Conclusion is made summarising most of the points taking into consideration the specifics of a given case and relevant recommendations are given. | The code provided is mostly correct and commented. Code has mostly proper use of variables, data types (list, tuple, dictionary or set), conditionals, loops, functions, modules. Code contains some errors |
| **50 – 59** | Adequate introduction and assumed needs. Review of software and hardware solutions and tools taking into consideration various range from referenced sources is present. Justification is neither too broad nor too narrow. Conclusion is partially made summarising the points taking into consideration the specifics of a given case and some relevant recommendations are given. | The code provided is mostly correct but poorly commented. Code lacks proper use of variables, data types (list, tuple, dictionary or set), conditionals, loops, functions, modules. Code contains some errors |
| **40 – 49** | General introduction and assumed needs. Some review of software and hardware solutions and tools taking into consideration some referenced sources is present. Justification is rather limited and could be enhanced. Conclusion is partially made summarising the points taking into consideration the specifics of a given case and a limited number of relevant recommendations are given. | The code provided contains errors and is not commented. There is a fair attempt to properly experiment with at least 3 different data types, conditionals, loop, function. |
| **30 – 39** | Brief introduction and assumed needs from business perspective. Limited review of software and hardware solutions and tools taking into consideration some referenced sources is present. Justification could be enhanced. Conclusion is rather general and limited | The code provided contains errors and is not commented. There is a limited attempt to experiment with at least 3 or less different data types, conditionals, loop, function. |
| **20 – 29** | Some description of the report without proper review of hardware review and software, justification | The code provided contains errors and is not commented. There is a very limited attempt to experiment with code |
| **10 – 19** | Extremely limited coverage of reviews of hardware of software, justification | The code provided contains errors and is not commented. There is an extremely limited attempt to experiment with code |
| **0 – 9** | Some attempt to perform introduction and conclusion. | Some attempt to experiment with code |