****

# Assessment Information/Brief 2021-22

|  |  |
| --- | --- |
| Module title | Computer Programming |
| CRN | 50162/50610 |
| Level | 3 |
| Assessment title | Assignment 2: Coding a survey app with a graphical user interface |
| Weighting within module | This assessment is worth 70% of the overall module mark. |
| Submission deadline date and time | Friday 17th December 2021 before 4pm For coursework assessments only: students with a Reasonable Adjustment Plan (RAP) should check their RAP to see if an extension to this submission date has been agreed. |
| Module Leader/Assessment set by | Dr Ian Drumm, i.drumm@salford.ac.uk |
| How to submit | You are to submit electronically via the Computer Programming **Blackboard** page….     * zipped folder containing your Python file and any resources it uses (e.g. images). The name of the folder and the name of the python file must be of the form… *surname\_initial\_application.* For example, a directory folder called **smith\_j\_application**, containing **smith\_j\_application.py** and accompanying images.   The assessors must be able to run this application straight from the command line without having to install any additional libraries or moving any resources. For example, by typing…  **python smith\_j\_application.py** |
| Assessment task details and instructions | Assessment task details and instructions  A giant ‘singing’ sculpture is about to tour the UK to promote the science of acoustics. The project coordinators and funders want to survey the people who viewed the sculpture.    To do this the funders require a Python application running on a nearby PC that viewers of the sculpture can use to fill in a form with their name, demographics (age, sex, ethnicity, disabled status, etc), and whether they a) enjoyed the sculpture, b) were curious as to how it worked and c) wanted to know more about science as a result.  Create an application with an attractive graphical user interface to enable viewers of the sculpture to complete and submit this form. The application should ideally be attractively presented and accessible e.g. with images, suitable colours and fonts, and a range of UI controls.  All data apart from the name must be amenable to a numerical representation for possible statistical evaluation, for example if people are asked if they enjoyed the sculpture, they could choose from 1 Strongly Agree, 2 Agree, 3 Neither Agree, 4 Disagree or 5 Strongly Disagree (i.e. a Likert scale). If asked ethnicity they can choose from 1 White, 2 Black, 3 Chinese, 4 Asian or 5 Other.  Ideally the application should include error checking, for example if a non-numerical age is entered how will you test for it and how will you deal with the error?  For higher marks the application should include a separate tab page or window so that the hosts of the application can view a list of the sculpture viewers’ names and the answers they submitted.  For higher marks the hosts should also be able to see statistical values associated with the data. For example, the average age, number of women who liked the sculpture, standard deviations, etc. The application could also put the hosts page behind a password as you wouldn’t want the sculpture viewers to see this.  The application should be written with the **Visual Studio Code** IDE, and all coding should be written in a **single Python file.** You should use **Tkinter** for the user interface. **Do NOT use third-party (‘pip installed’) libraries**. All application files and resources must be in the same directory, **and the application must be runnable from the command prompt or terminal of any PC with Python installed**.  The code should also be well commented with the comments explaining what key lines or sections of code do with appropriately clear, concise and technical language. Comments are an opportunity to demonstrate one’s understanding of the code and one’s ability to use the correct terminology. |
| Knowledge and Understanding  Practical, Professional or Subject Specific Skills | Assessed intended learning outcomes  On successful completion of this assessment, you will be able to:  1. create structured object-oriented programs,  2. analyse problems, design solutions, discuss alternatives and test results,  1. program in an industry standard language using an industry standard IDE,  2. create simple desktop applications and web applications to address well-defined real-world problems. |
| Module Aims | 1. to introduce computer programming using object-oriented principles, 2. to develop simple applications that do useful real-world things. |
| Word count/ duration (if applicable) | All features of the application should be usable within 5 minutes of running the application. |
| Feedback arrangements | You can expect to receive feedback within 15 student working days. Each student will also have written feedback broken down according to the marking criteria. |
| Support arrangements | After week 10, formal teaching of new topics will have finished. The subsequent scheduled classes will be surgeries to support students towards getting started with their assignments. |
| askUS  Good Academic Conduct and Academic Misconduct | The University offers a range of support services for students through [askUS](http://www.askus.salford.ac.uk/).  Students are expected to learn and demonstrate skills associated with good academic conduct (academic integrity). Good academic conduct includes the use of clear and correct referencing of source materials. Here is a link to where you can find out more about the skills which students need <https://www.salford.ac.uk/library/skills-for-learning>.  **Academic Misconduct is an action which may give you an unfair advantage in your academic work. This includes plagiarism, asking someone else to write your assessment for you or taking notes into an exam. The University takes all forms of academic misconduct seriously. You can find out how to avoid academic misconduct** [**here**](https://www.salford.ac.uk/library/skills-for-learning/academic-skills/making-most-feedback)**.** |
| Assessment Information  Personal Mitigating Circumstances  Personal Tutor/Student Progression Administrator | If you have any questions about assessment rules, you can find further information in Blackboard in the Assessment Support area.  If personal mitigating circumstances may have affected your ability to complete this assessment, you can find more information about personal mitigating circumstances procedure [here](https://www.salford.ac.uk/askus/admin-essentials/personal-mitigating-circumstances).  If you have any concerns about your studies, contact your Personal Tutor or your Student Progression Administrator. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assessment Criteria | The following is a guide to the criteria that you need to satisfy in order to get a grade within each of the following ranges.   |  | **80-100%** | **70-79%** | **60-69%** | **50-59%** | **40-49%** | **Below 40%** | | --- | --- | --- | --- | --- | --- | --- | | **User Experience**  **25%** | Outstanding user experience, relative to other submissions, which fits the brief perfectly. | Is particularly elegant, intuitive, accessible and pretty with an adventurous ranges of UI controls going beyond the module notes.  Expands functionality without need for external libraries and goes beyond the module notes, for example, includes file io, graphs,, advanced stats, etc. | An enjoyable and appropriate user experience fitting the assignment brief.  Good functionality, such as a host’s page with a password and a good range of stats. | Competent user experience.  Is simple and intuitive. Though not as pretty or accessible as it could be.  Robust with error checking.  Considers two use cases, e.g. a survey participant page showing the survey form and a hosts page showing survey results. | An adequate but limited user experience, perhaps lacking in simplicity, ease of use or attractive presentation.  Adequate but incomplete functionality, e.g. a survey participants page showing the survey form but no hosts page showing survey results.  Lacking some error checking of input. | Very poor user experience.  Doesn’t fit the brief.  Application doesn’t run directly from the command line. | | **Coding**  **75%** | Outstanding functionality  The use of variable names, function names, etc are impeccable.  The use of comments shows a mastery of Python and OOP concepts presented in the module. | Very good, elegant coding.  Very good presentation of coding with appropriate comments, naming conventions, indenting, etc.  Clearly building on notes and exercises presented throughout the module. But with exploration of wider features available without ‘pip’ installing additional libraries. | Good creative programming with impressive functionality.  Thoughtful, tidy and elegant coding.  Comments show a good technical understanding of Python features and OOP employed. | Competent application with basic functionality fitting the assignment brief. This includes the application of class definition and instantiating objects to manage with a list. Making use of examples in notes.  Code tidy with comments showing a fair understanding of Python employed.  Student has demonstrated he/she has taken responsibility for own learning. | Adequate, working code though limited functionality.  Not fitting the brief as well as could be.  Lacking real thought of design.  Sloppy, untidy coding poorly presented.  Sparse or excessive use of comments. | Not a fully working application  Very untidy presentation makes code  difficult to read and understand.  Functionality doesn’t meet with the requirements of the assignment briefing.  No meaningful engagement with Python programming. | |

|  |  |
| --- | --- |
| Percentage Mark Level of Performance  90-100 Outstanding  80-89 Excellent  70-79 Very Good  60-69 Good  50-59 Fair  40-49 Adequate  30-39 Unsatisfactory  20-29 Poor  10-19 Very Poor  0-9 Extremely Poor |  |

|  |  |
| --- | --- |
| In Year Retrieval Scheme | Your assessment is eligible for in year retrieval. If you are eligible for this scheme, you will be contacted shortly after the feedback deadline. |

|  |  |
| --- | --- |
| Reassessment | If you fail your assessment, and are eligible for reassessment, you will need to resubmit on or before July 31st 2022. For students with accepted personal mitigating circumstances for absence/non submission, this will be your replacement assessment attempt.  The reassessment task will be the same. |