Coursework or Assessment Specification

Module Details

Module Code	**UFCFVW-6-1**
Module Title	**Object-Oriented Programming (Course Project)**
Module Leader	**Boriss Mišņevs**
Module Tutors	**Aleksejs Vesjolijs**
Year	2022/23
Component/Element Number	PROJ1
Total number of assessments for this module	1
Weighting	**100%**
Total Assignment Time	
Element Description	Course project

Dates

Date issued to students	15/02/2025
Submission Date	31/04/2025
Submission Place	Moodle
Submission Time	23:59
Submission Notes	The report and application should be submitted through Moodle before the due date

Feedback

Feedback provision will be	Feedback will be provided in the Moodle environment regarding each item.

Contents

Section 1: Overview of Assessment

The Course project is a logical continuation of the Object-Oriented Programming module. This assignment is targeted at the development of the professional skills and applying knowledge received during the Object-Oriented Programming module.

This assignment assesses the following module learning outcomes:

Module Learning Outcomes	**Reference**
Understand program lifecycle and program development stages	MO1
Use user interface development	MO2
Use object-orientated paradigm for software development	MO3
Use proper coding style, considering OOP paradigm	MO4
Apply basics of documenting software products	MO5
Use of development environments and debuggers for program creation and testing purposes	MO6
Use of development environment	MO7

All work for this module is individual and should be completed **individually**.

Broadly speaking for this module, you are required to:

1. design, develop, and test an application for a given task
2. submit all source code and all repositories
3. write a technical report.

All work should be uploaded to the Moodle course which can be found at https://e.tsi.lv/course/view.php?id=2586

If you have questions about the assignment, **ask your course leader.**

Section 2: Task for programming course project

INTRODUCTION

The Course project is a logical continuation of the Object-Oriented Programming module. This assignment is targeted at the development of the professional skills and applying knowledge received during the Object-Oriented Programming module. Working on the Course project, you will have the ability to apply the following knowledge and skills that you acquired during the Object-Oriented Programming module:

- use of syntax of the C# or Java programming language;
- application of OOP programming paradigm;
- create and use user-defined data types, such as classes and interfaces;
- application of inheritance, polymorphism and encapsulation
- development of a user graphical interface;
- exception handling;
- serialization of objects:
- storing and processing information in files or in database;
- software development stages.

1. GOAL

The main goal of the Course project is to develop an application with graphical user interface using Object-Oriented Programming language C# or Java. You can choose the programming language and type of the application

- Desktop application
    - using C# programming language: Windows Form Application or Windows Presentation Foundation (WPF) application; or
    - using Java programming language and JavaFX

or

- Web application
    - using C# programming language and ASP.Net; or
    - using Java programming language and Spring.

The general functionality of the application is to process (add, delete, update, filter sorts), store, and display data on the screen.

Alternatively, you can create back-end web-application or terminal computer program **without** user interface, but it must be done in Object-Oriented Programming language C# or Java and written by OOP standards, also in this case complexity requirements for application is increased.

- Server-side / Back-end application
    - using C# programming language and ASP.Net
    - using Java programming language
- CLI (Terminal) Computer Program
    - using C# programming language
    - using Java programming language

2. THE THEME OF COURSE PROJECT

Each student should complete the Course project **individually** on a topic area selected by themselves. The specific title of the Course project depends on the selected application area. An example title could be something like: *Development of the information system for library service*. Other examples of subject areas can be found in the *Section 5*.

3. THE MAIN TASK

The main task of the Course project is to design, develop and test an application that will process and store information for a specific subject area, of your choice. Three deliverables are expected: application, source code and report.

4. GENERAL REQUIREMENTS

Requirements for the Course project are presented below:

- The application should fulfil all the requirements set for it.
- Object-Oriented approach should be used for implementation.
- The program should work without critical errors.
- The program should have a clear user interface.
- The report should contain items, which demonstrates the design and development process of the application. *See Section 7 for details.*
- The report design should follow the requirements. *See Section 8 for details.*
- Data about objects should be declared as **user-defined types** - classes
- Data should be stored in files in any formats (binary, XML, JSON)

5. TASKS

1) Analysis of the subject area.
    a. Select a subject area of your choice and begin to analyse it, identify the objects (entities) which should be considered by the application. *The examples of subject areas see in the Table below*.

The examples of subject areas

Nr	Subject Area	Description
1	Bank Management System	Maintain the record and all banking related tasks in this project
2	Fin-Tech Application	Software Application for fin-tech operations
3	Hackathon Management System	Application for Hackathon organization
4	Startup Crowdfunding Network	Social network application used to fund startups
5	Video-editor application	Capture, edit and save videos
6	Radio Centre	Connect & listen to various radio stations
7	P2P Messenger	Online messenger app
8	Master for an hour	Call control system for minor repairs
9	Museum	Catalogue of art
10	Bicycle rental	Control the location of bicycles and create routes for their collection
11	Social network for animals	Organization of partners for pets
12	Cleaning company	Preparing routes and schedules for home cleaning
13	Chocolate factory	Control of supplies to stores
14	Hydra (from comics)	Spy monitoring and information collection system
15	Mystical bureau	Unexplained facts registration system
16	Construction company	House building monitoring system

    b. Identify the characteristics (attributes) of the objects (entities). **The minimum number of each object's attributes is 3**. Define the data that will be stored in files, define data types, restrictions, etc.
    c. The application should be able to operate with **minimum 3 types of objects. Note! The number of objects and consideration of interrelation between objects has a direct impact on grading**, *See section 9 for additional information.*
2) Describe functional requirements. Determine and describe the functions that the system should provide. The obligatory functions of the system should be:
    a. adding and storing the data about objects in the **files**,
    b. data display (one object, several objects, in the form of a table),
    c. deletion of data about objects (deletion according to the specified criteria),
    d. search and filtering of data (the ability to set different criteria, at **least 3 criteria**),
    e. calculation of any summary characteristics, for example, the number of records with a certain value of any field (at least **2 characteristics**),

f. sorting (according to several criteria **at least 2),**

3) Define attributes that will be used for search, filtering, deleting, sorting and summary.
4) Develop user-defined data types - classes for each object. Each identified object should have its own class. Consider additional data types: enum, interfaces etc. When designing classes, keep in mind that:
    - private fields,
    - properties,
    - abstract classes,
    - inheritance,
    - interfaces

should be used. **The use of these concepts affects the final grade.** *See section 9 for additional information.*

Develop the class diagram.

5) Develop a user interface, which will allow completing earlier identified functions with the objects.
6) Develop algorithms that perform the identified functions in form of flow diagrams.
7) Implement a program code in C# or Java programming language.
8) Test the program.
9) Prepare the report about completed work.

6. RECOMMENDED STAGES OF THE WORK

It is recommended to perform the Course project in stages. The stages of the project are the following:

1) **Formulation of the problem.** At this stage, it is necessary to draw up a list of requirements for the information system, which should include.
    a. Subject area description
    b. Objects and their attributes description
    c. List of functional requirements with an indication of the input data for each operation and the expected result**.**

2) **Design.** At this stage it is necessary to develop the structure of the program:
    a. Design of user-defined data types (classes, interfaces, enum), class diagram.
    b. Design of user interface (structure of the menu, forms, or pages layouts)
    c. Develop algorithms for methods. Algorithms should be presented as flowcharts. Flowcharts could be commented in the text if it is necessary.

3) **Development.** Using C# or Java programming language define designed user-defined data types (classes, interfaces etc). Each class should be defined in a separate source file. From forms or pages layouts develop the graphical user interface for your application. Connect user interface with developed classes. The output of this stage is a working application, which could have errors, but at the same time able to perform core operations.

4) **Testing and debugging.** At this stage, it is necessary to develop a set of tests for the software. Run the program with the specified tests, and fix errors that could be found. At the end of the stage, there should be a program in which the maximum possible errors were found and corrected. A description of the testing process is expected in the report.
5) **Report preparation.** At this stage, the documentation is drawn up following the requirements for the preparation of the report. *See Section 7 and Template of Report.*
6) **Submit code to the GitLab.** Final code should be uploaded to GitHub repository, created specially for the Course Project. It might be both ***public*** or ***private*** repository.

7. CONTENT OF REPORT

Below is presented the recommended structure of the report, which should be prepared as the main deliverable of the Course Project. **!Note that some parts of the report are considered in the grading scheme as separate elements.**

Title page *– use the template provided in Moodle.*
Content – *should be generated automatically and should reflect all sections and subsections of the report.*
Introduction – *this section should introduce the reader to the task you are implementing, introduces very briefly the subject area. Briefly describes each chapter (section of the report). Recommended size – max 1 page.*
1. Tasks for the Course project – *describes the tasks of the Course project in a detailed way. Recommended size – max 1 page.*
2. Analysis of the subject area *– this section should include the information about the selected subject area, an overview of the subject area, identification of the core entities (objects), their attributes (characteristics), the interrelation between objects. Discuss and argue, what attributes should be used for sorting, searching etc. We are also welcome here not only text but figures, tables etc.*
3. Design *– this section should include information about the design of the application. Provide here your vision of how navigation will be organised (menu design), what user types you will use to describe entities (objects). Provide information about files that will store the data. Present here class diagram, algorithms as flowcharts for the methods you are planning to implement, forma or pages layouts. We are also welcome here not only text but figures, tables etc.*
4. Development *– provide in this section information about the development. Describe in details, created user-defined types, developed user interface, list of source files. We are recommending utilising tables. For example, a class could be described in the following way: fields with a type and a description, properties with a type and a description, methods with a signature, description of the result and of the arguments.*

Development section should contain link to the GitHub.

Section should contain information about the developed application and its functionality. Demonstrate screenshots of the developed applications and results of completing different operations. We are also welcome video records here with the demonstration.

6. Testing *– provide here information about how you have organised the testing of the developed software. List here scripts, scenarios, test cases, test results, details about found errors and comments, how errors have been corrected or not corrected.*

7. Conclusions *– Indicate in this section what has been done in the frame of the course project. Sum up information about the application and development process. Briefly describe the main functionality of the software. Indicate the main issues during the development process. Complete the analysis of your application, what has been done well, what could be improved, what has caused most of the problems during design, development, testing etc.*

8. References *– any references you have used.*

Annexe: *source code.*

8. FORMATTING YOUR REPORT

Report design requirements that should be considered preparing the report:

- **Pages:**
  - **Page size:** A4
  - **Margins** - left - 3 cm; right, top, bottom - 2 cm
  - **Page numbering** – at the bottom of the page in the middle, the title page should not be enumerated.
- **Heading**:
  - 1st level - 14 pt, Times New Roman, Bold, Upper Case, alignment: centre
  - 2nd level – 12pt, Times New Roman, Bold, Sentence Case, alignment: left
  - 3rd level – 12pt, Times New Roman, Italic, Sentence Case, alignment: left
- **Main text:**
  - Font Times New Roman, size 12pt, line spacing 1.5.
  - First-line indent - 1 cm.
  - Alignment – justify.
- **Figures/Pictures:**
  - **Caption template**: Fig. <number> Title of the figure
  - **Caption text:** Font Times New Roman, size 10pt
  - **Caption location:** Under figure, centred
- **Tables**:
  - **Caption template**: Table <number> Title of the table
  - **Caption text**: Font Times New Roman, size 10pt
  - **Caption location**: Above the table, centred
- **Sections** – each section (chapter) starts on a new page.
- **Reference** – if any, we are recommending using Harvard style.
- **Application (code)** – font-size: 8 font size, line spacing – 1

See the Report template in the specific Moodle section as an example.

9. GRADING SCALE

We are strongly recommending making yourself familiar with the **grading scheme** presented below.

An application, files with source code and a report submission required!

Maximum number of points **-220 points**
Passing points amount – **88 points.**

	Criteria	***Marks***						
1	Report design	***10***	The outstanding design of the report. All requirements are fulfilled, the report has a uniform design. ***9-10 points***	The design of the report follows the requirements, but some minor issues are observed. ***7-8 points***	The report in general follows the requirements but has a significant number of minor issues in the design. ***5-6 points***	The report attempts to follow the design requirements. But major issues are observed, they make report partly ununiformly designed. ***3-4 points***	The report is designed in ununiform style, but it does not follow the requirements. ***1-2 points***	The report does not follow the design requirements at all. It has an ununiform design. Or the report is not submitted. ***0 points***
2	Analysis of the subject area	***10***	Outstanding description of the subject area. The analysis of the subject area is completed, results are clear and well presented. Requirements are fully covered. ***9-10 points***	The description is clear and covers all the required points. The analysis is completed, and analysis of the results are presented clearly. But some extension could be done. ***7-8 points***	A moderate description is provided. It gives a view on the subject area. A light analysis of the subject area is provided. ***5-6 points***	The description is provided in general form, but at the same time attempts to complete the analysis is foreseen. ***3-4 points***	The description is presented in a minimal volume. It is too generic and does not provide a clear vision on the subject area. No attempts to complete the analysis. ***1-2 points***	The description is not presented, or the description does not provide a clear vision on the subject area. ***0 points***
3	Design section of the report	***10***	Outstanding presentation of the design section of the report. The provided information is clear and does not require additional comments/extension/clarification . The section has an excellent relationship with the section Analysis of the subject area. ***9-10 points***	The report has a section devoted to the design. The provided information is clear has a good relationship with the information provided in section Analysis of the subject area. All requirements are well covered, only minor issues are identified which requires additional clarification. ***7-8 points***	The report part devoted to the design is presented in the report. The presented information is moderate, but it the same time it gives useful and understandable information about the future application. There is a significant number of minor issues, which could be extended/clarified/described to make this section clear. Moderate relation with the section Analysis of the subject area is observed. ***5-6 points***	The report part devoted to the design is presented in the report. Provided information gives a general vision on the design of the application. The information corresponds in general to the requirements, but the significant number of gaps in the description does not allow to get a clear understanding of the information. Moderate relation with the section Analysis of the subject area is observed. ***3-4 points***	The report has a section devoted to the design. But provided information is limited or partly covers requirements. It does not provide a clear vision on the design of the application. Provided information has a partial relationship with the section Analysis of the subject area. ***1-2 points***	The report does not include information about application design. The section is missing or provided information is out of scope. ***0 points***
4	The developme nt section of the report	***30***	Outstanding presentation of the development section of the report. The provided information is clear and does not require additional comments/extension/clarification . Link to GitHub is present ***25-30 points***	The report has a section devoted to development. All requirements are well covered, only minor issues are identified which requires additional clarification. Link to GitHub is present ***19-24 points***	The report part devoted to the development is presented in the report. The presented information is moderate, but it the same time it gives useful and understandable information about the development. There is a significant number of minor issues, which could be extended/clarified/described to make this section clear. ***13-18 points***	The report part devoted to the development is presented in the report. Provided information gives a general vision on the development of the application. The information corresponds in general to the requirements, but the significant number of gaps in the description does not allow to get a clear understanding of the information. ***7-12 points***	The report has a section devoted to development. But provided information is limited or partly covers requirements. It does not provide a clear vision on development. ***1-6 points***	The section is missing or provided information is out of the scope. ***0 points***

	Criteria	***Marks***						
5	Testing section of the report	***10***	The report includes clear information about application testing, test cases are well described. Testing results are presented in clear form. Errors correction actions are described. All functions are tested, also boundary inputs are considered. ***9-10 points***	The report has a section devoted to testing. The provided information is clear and meaningful. All main function has been tested, errors were identified, information has about their correction has been provided and they are clear. The attempt to consider boundary inputs in test cases has been done and information provided. ***7-8 points***	A report section about testing is presented. The provided information is moderate, but all main functions of the application are covered. Errors indicated, but information about their correction is a moderate one. In test cases, boundary inputs are not considered. ***5-6 points***	The report part devoted to the testing is presented in the report. Provided information gives a vision on how testing has been conducted. Test cases are provided. The majority of the main functions are tested, but not all. Errors are indicated, but information about how they have been corrected is limited. ***3-4 points***	The report has a section devoted to the testing. But the provided information is limited. It does not provide a clear vision on test cases. Few main functions are tested. The errors are indicated, but no information about how they have been corrected. ***1-2 points***	The report does not include information about application testing. The section is missing or provided information is out of scope. ***0 points***
6	Conclusions	***20***	Outstanding conclusion. They sum up information about the developed application and development process. They demonstrate analysis results and emphasise the issues. Provides critics about development selected solutions etc. ***17-20 points***	Good conclusions, which sum up information about the developed application and development process. The attempt to complete the analysis and present analysis results at a good level. ***13-16 points***	Moderate conclusions. They demonstrate what has been done in details. Moderate attempt to make analysis and indicate any issues during development. ***9-12 points***	Most of the conclusions are too general. Conclusions present what has been done in a generic form. No attempt to make an analysis and indicate any issues during development. ***5-8 points***	Conclusions are very light. They are not clear, do not demonstrate what has been done in the frame of the project. No attempt to make an analysis and indicate any issues during development. ***1-4 points***	The section is missing or provided information is out of the scope. ***0 points***
7	The complexity of the application	***40***	Count of objects 3 or greater Count of attributes >=9 Properties, an abstract class, an inheritance, interface are provided. The relationship between objects is implemented. All obligatory functions (6) have been implemented. Check for the correctness of user input is implemented. ***34-40 points***	Count of objects = 3 Count of attributes <9 Properties, an abstract class, an inheritance, interface are provided partly. The relationship between objects is not implemented or implemented partly. All obligatory functions (6) have been implemented. ***27-33 points***	Count of objects = 2 Count of attributes >=6 Properties, an abstract class, an inheritance, interface are provided partly. The relationship between objects is implemented. All obligatory functions (6) have been implemented, but not for all criteria. ***20-26 points***	Count of objects = 2 Count of attributes <6 Properties, an abstract class, an inheritance, interface are provided partly. The relationship between objects is not implemented. Count of implemented obligatory functions 4-5 ***13-19 points***	Count of objects = 1 Count of attributes >=3 Properties, an abstract class, an inheritance, interface are not provided. Count of implemented obligatory functions 2-3 ***6-12 points***	Count of the objects =1 Count of attributes <3 Properties, an abstract class, an inheritance, interface are not provided. Count of implemented obligatory functions =1 ***0-5 points***
8	*Code*	***30***	The code is readable, understandable, contains meaningful comments, formatted according to the recommendations, the names of the user-defined types, methods, variables correspond to the recommendations. Method overloading has been implemented. The code is logically divided into several source files. ***25-30 points***	The code is readable, understandable, contains moderate comments, formatted according to the recommendations, the names of the user-defined types, methods, variables correspond to the recommendations. Method overloading has been implemented. The code is logically divided into several source files. ***19-24 points***	The code is readable, understandable, contains moderate comments, formatted according to the recommendations, the names of the user-defined types, methods, variables are meaningful. Method overloading are not used. The code is not split into several source files. ***13-18 points***	The code is readable, understandable, contains light comments, formatted according to the recommendations, the names of the user-defined types, methods, variables are meaningful. Method overloading are not used. The code is solid, not logically divided. ***7-12 points***	Code is developed, but code is unreadable, or not formatted, does not have comments, or the names of the user-defined types, methods, variables are not meaningful. The code is solid, not logically divided. ***1-6 points***	The code is missing. ***0 points***

	Criteria	***Marks***						
9	User interface (Option 1)	***30***	The application user interface is developed. There are more than 5 views (forms, pages). All inputs and outputs have explanatory plain text. The hints are implemented. A menu is implemented. The input is checked for correctness. The forms or pages work without errors. ***25-30 points***	The application user interface is developed. There are 4-5 views (forms, pages). All inputs and outputs have explanatory plain text. The menu is implemented. Checking for the correctness of input is not performed or the hints are not implemented. The forms or pages work correctly. ***19-24 points***	The application user interface is developed, but there are some flaws in the explanatory texts for input and output. There are 2-3 views (forms, pages). The menu is implemented. Checking for the correctness of input is not performed. The hints are not implemented. The forms or pages work correctly in most cases. ***13-18 points***	The application user interface is developed, but there are some flaws in the explanatory texts for input and output. There are 1-2 views (forms, pages). The menu is not implemented. Checking for the correctness of input is not performed. The hints are not implemented. The forms or pages work correctly in most cases. ***7-12 points***	Application has unclear user interface – user cannot understand what application waits for inputs and what the application outputs as a result. There are only 1 view (form, page). The menu has a minimum number of options for the user. ***1-6 points***	The application user interface is not developed. ***0 points***
10	Additional complexity of the application without user interface (Option 2)	***30***	Count of objects 7 or greater Count of attributes >=15 Properties, an abstract class, an inheritance, interface are provided. Minimum 5 methods. The relationship between objects is implemented. All obligatory functions (6) have been implemented. Check for the correctness of user input is implemented. ***25-30 points***	Count of objects = 6 Count of attributes <12 Properties, an abstract class, an inheritance, interface are provided partly. Minimum 4 methods. The relationship between objects is implemented. All obligatory functions (6) have been implemented. ***19-24 points***	Count of objects = 5 Count of attributes >=10 Properties, an abstract class, an inheritance, interface are provided partly. Minimum 3 methods. The relationship between objects is not implemented or implemented partly. All obligatory functions (6) have been implemented, but not for all criteria. ***13-18 points***	Count of objects = 4 Count of attributes >10 Properties, an abstract class, an inheritance, interface are provided partly. Minimum 2 methods. The relationship between objects is not implemented. Count of implemented obligatory functions 4-5 ***7-12 points***	Count of objects = 3 Count of attributes >=10 Properties, an abstract class, an inheritance, interface are not provided. Minimum 1 method. Count of implemented obligatory functions 2-3 ***1-6 points***	Count of the objects <3 Count of attributes <9 Properties, an abstract class, an inheritance, interface are not provided. Count of implemented obligatory functions =1 ***0 points***