# Simple Music Composer using Music Keyboard and Music Staff Components

## Purpose:

To apply Object Oriented Programming technique supporting code reusability and team development

## I. SPECIFICATION.

**General Objective:** A Window app producing music score/script displayed on screen by playing a ***Music keyboard*** with help of mouse clicks. The music score consists of ***Music Staff*** object populated by a played melody in form of ***Music Notes*** objects/components.

### Features of the *MusicNote* Class (Component):

- *data fields*:
  - pitch (integer number which maps to the corresponding .au or .wav file)
  - note shape (*semi-breve, dotted-minim, minim, crotchet, quaver, semi-quaver ...as* .BMP files)
  - note duration (duration in milliseconds determining a corresponding shape of music note)
  - accidental: enumerated type (flat, sharp, sole)
    E.g. enum Accid {flat, sharp, sole}; Accid ac = Accid.flat; int num = (int) ac ;
- ***mouse events*** (when the mouse pointer is positioned on a particular music note image):
  - *Click* event: plays a single note of the selected music note with its duration and pitch
  - *RightPress* event: used for editing of of the selected music note. The duration of the right mouse button pressed determines the shape of music note. When released the duration value is set accordingly.
  - *Drag* event: moves the existing music note vertically only and adjusts its pitch accordingly.

### Features of the *Music Staff* Class (Component):

- ***A subclass of the Panel*** component. When constructed in the main window it displays itself as a rectangle of a fixed size with 5 lines of a music staff. The lines are fixed relative to the upper left corner of the panel
- *data fields:*
  - a collection storing *Music Note* objects as they were produced by playing the keyboard
  - a "play" button: plays the music melody as stored on the collection (and displayed on the music staff)
  - a tempo field: integer reflecting overall speed in ms (allegro / adagio, etc)
  - a "save" button storing the collection of the melody on a file stream
  - a "load" button loading the collection of the melody from the file stream
- *operations*
  - playing of the individual music note displayed on the music staff manually by pointing and clicking mouse LB
  - playing of the music notes continuously by clicking on the "play" button (plays the whole melody by traversing the collection of music notes)
  - adjusting the overall tempo by setting the tempo field (*grave, largo, lento, adagio, andante, moderato, allegro, presto*)

### Overall activity:

- At the beginning both the *Music Staff* and *Music Keyboard* objects (components) are constructed and displayed
- *Music Note* object is constructed and displayed when a particular keyboard key is pressed by mouse. The duration of keeping the key pressed determines the type (shape) of the music note generated.
- By dragging the *Music Note* into a new position on the Music Staff its *pitch* property is changed accordingly
- Mouse LB click plays the individual *Music Note* pointed to on the staff
- Mouse RB click changes duration of the individual *Music Note* on the staff and its shape accordingly
- By pressing the "play" button the music melody is played by traversing the collection and in tempo dictated by the Progress Bar

## II. FORMAL ASPECTS

1. **Team work**
   Team will consist of two students, with the following responsibility:
   ***Person 1***: design and implementation of music note class including all associated events and related documentation.
   ***Person 2***: design and implementation of music staff and music keyboard classes including their associated events and related documentation
2. **Documentation**
   The documentation (one copy for both partners) will have a character of working document reflecting the way of information exchange between the team of program designers. As such it will be brief, but information rich.

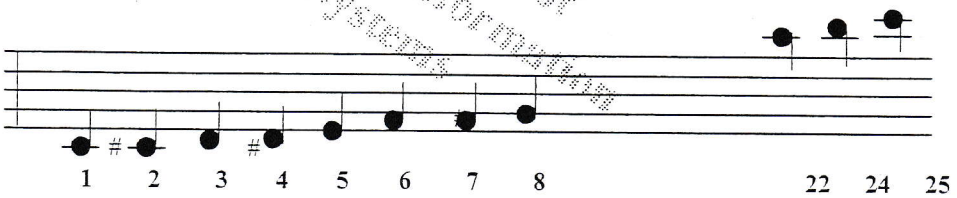
1) Front page: Title of project, Names of authors, Study Unit specification (CIS 2101), year 2018/2019
2) Specification: printout of the page number 1 of this document
3) Specification of the work stating class(es) and events developed by each member of the group
4) Design (class hierarchy, plus other important and relevant design issues)
   Full description of :
   - Process of converting music note bmp image to be transparent (not covering lines of staff by rectangle)
   - Finding precise/unambiguous location of the music note on the staff when dragging is stopped (should cover also correct locations for # marked notes)
   - Algorithms/solutions of more complex cases (e.g. implementation of tempo, placing accidentals in front of music note, etc.)
5) Implementation
   - Description of **core** (relevant) methods in form of algorithm and reference to a source code in documentation listing
   - Source code listing preceded with **Content in alphabetical order** of method names with reference to the page no of their occurrences in the listing
6) Evaluation
   Stating up to which level is the project completed, suggestions to extend, complete etc.

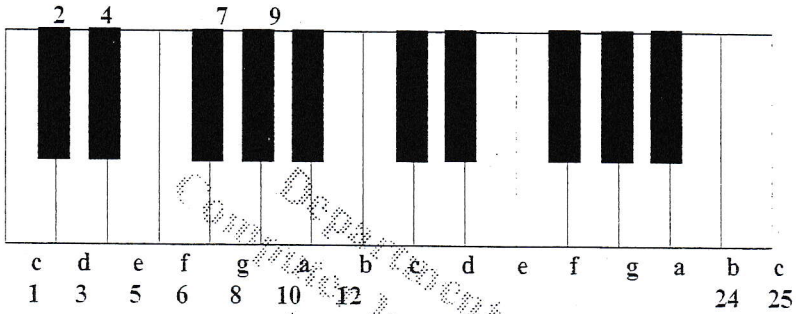
3. **Date of Submission and Demonstration**
   The latest submission is just 1 day before the examination od study unit CIS2201. Demonstration is of working program, duration 10-15 min. The date and time of demonstration will be announced.
4. **Assessment** The report, compiled programs, user manual and presentation will count for 40%, 40%, 10% and 10% of the total mark respectively.



**Music Staff**

**Details**




**Shapes of Music Notes**

¼ T ~ 63 ms ~ **one** timer tick (Timer.Interval = 63)

- semi-breve, duration 4T (1024ms ~ **16-20** ticks)
- dotted minim, duration 3T (768 ms ~ **11-15** ticks)
- minim, duration 2T (512 ms ~ **6-10** ticks)
- crotchet, duration 1T (256 ms ~ **3-5** ticks)
- quaver, duration ½ T (128 ms ~ **2** ticks)
- semi-quaver, duration ¼ T (63 ms ~**1** tick)