# Project 3 (20%)

## Problem domain: Priority Queues

The objective of this project is to enable you to apply the concepts of data structure design and implementation. You are required to do the following tasks:

1. Design data structure(s) for the given problem domain.
2. Devise a set of algorithms using the Algorithmic Definition Language (ADL) to manipulate the data structures designed in (i).
3. Implement the algorithms you devised in (2) systematically in either C# or Java as a console application and test your implementation.
4. A video presentation that showcases the capabilities of software in (2).
5. A descriptive report on any aspect of design and development, including artefacts produced for the project.
6. Produce a short report outlining response to formative feedback.
7. Add your solutions to your portfolio using the portfolio template provided.

This project represents 20% of the total marks available in this module.

**Note that the deadline for completion of this project is the same as the submission date of your portfolio. However, you are advised to complete it by 8th November 2021 (i.e., Week 16) to avoid increasing your workload as other projects will be given out throughout the semester.**

### Priority Queue

A *priority queue* is a collection of elements such that each element has been assigned a priority in which elements are deleted and processed comes from the following rules:

1. An element of higher priority is processed before any element of lower priority
2. Two elements with the same priority are processed according to the order in which they were added to the queue.

An example of the use of a priority queue is in a time-sharing operating system: programs of high priority are processed first, and programs with the same priority form a standard queue. There are various ways of maintaining a priority queue in memory. In this project, you are expected to focus on one that uses a linked list. Specifically, in this project, design and implement a priority queue for “job elements” as shown below. Note that the jobs are not in a particular order in this example. In your data structure design, you are required show the beginning and end of the priority queue.

|  |  |
| --- | --- |
| **Job Description** | **Priority** |
| EEE | 4 |
| GGG | 5 |
| CCC | 2 |
| DDD | 4 |
| BBB | 2 |
| FFF | 4 |
| AAA | 1 |

As outlined in the objectives, devise algorithms in ADL as suitable abstractions and implement them that add a job element (i.e., its description and priority), remove a job element and retrieve the priority of a job element given its description. Finally, implement your algorithms in either C# or Java as a console application.

This project will be graded using the criteria given below.

|  |  |
| --- | --- |
| **Criteria** | **Marks** |
| An assessment of understanding and use of algorithmic notation | 15 |
| An assessment of understanding of algorithm design | 30 |
| An assessment of data structure design | 30 |
| An assessment of overall software, including testing | 20 |
| Incorporation of formative feedback | 5 |