10 LABS x 2 hours 15 minutes

- Lab 1: OOP Reviews & Arrays
- Lab 2: Simple sorting
- Lab 3: Stacks & Queues
- Lab 4: Linked List
- Lab 5: Recursion
- Lab 6: Trees
- Lab 7: Hash Tables
- Lab 8: Graph
- Lab 9: Exam
- Lab 10: Project Presentation

There are 8 practical labs (30%):

- Select 3 random submissions to mark
- If you miss a lab or a submission: that lab will be selected to mark

Lab 9 will be a practical exam (35%)

- You can use your laptop to code
- You are only allow to use the following IDE:
    - NetBeans
    - VS Code
    - BlueJ
    - IntelliJ
- You must DISCONNECT your laptop from the Internet

Lab 10 is the project presentation (35%)

Deadline to submit your work on Blackboard: 3 days from the lab day

- i.e., Lab day is Monday => deadline is Wednesday (mid-night)

Assignments submission guide

- Create the folder with a name like: ***StudentID_Name_Lab#***, (e.g. ***01245_VCThanh_Lab1***) to contain your assignment with subfolders:
    - Problem_01 (sometimes Problem_i or Problem_Array)
    - Problem_02 (sometimes Problem_ii or Problem_Queue)
    - etc.
- Compress (.zip) and Submit the whole folder with the same name (i.e., ***01245_ VCThanh_Lab1.zip***) to Blackboard
- Students **not** following this rule **will get their marks deducted**

3. Lab 3: Stacks & Queues

3.1. Objectives

- Know how to use the data structure Stack for solving real problems.

3.2. Problem 1: Simple stack application

Write a program to

- Convert a decimal number and convert it to octal form.
- Concatenate two stacks.
- Determine if the contents of one stack are identical to that of another.

3.3. Problem 3: Undo and redo

Write a class *SpecialArray* that has:

- (8 points) an array of 20 random values
- (8 points) a function to update the value at a position in the array.
- (8 points) a function to undo the updating.
- (8 points) a function to redo the updating.
- (8 points) a function to display content of the array.

Hint: use two stacks to store the array after each operation

3.4. QueueApp.java

- Write a method to display the queue array and the front and rear indices. Explain how wraparound works.
- Write a method to display the queue (loop from 1 to nItems and use a temporary front for wraparound).
- Display the aray, the queue, and the front and rear indices.
- Insert fewer items or remove fewer items and investigate what happens when the queue is empty or full.
- Extend the insert and remove methods to deal with a full and empty queue.
- Add processing time to the queue. Create a new remove method that removes item N after N calls to the method.
- Simulate a queue of customers each one served for a random amount of time. Investigate how simulation is affected by:
    - the size of the queue
    - the range of time for wich each customer is served
    - the rate at which customers arrive at the queue

3.5. ReverseApp.java

- Create a stack of objects of class Person and use to reverse a list of persons.

3.6. PriorityQApp.java

- Write a method to display the queue and use to trace the queue operation.
- Modify the insert method to insert the new item at the rear. Compare this queue with QueueApp.java. Which one is more efficient?
- Use a priority queue instead of an ordinary one in the simulation experiments described above.