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**

5. Lab 5: Recursion

5.1. Objectives

Apply Recursion in pracitical mathematical problems.

5.2. Problems

5.3. Problem 1: Use the following function puzzle(..) to answer problems 1 - 3.

```
int puzzle(int base, int limit)
{       //base and limit are nonnegative numbers
        if ( base > limit )
                return -1;
        else if ( base == limit )
                return 1;
        else
                return base * puzzle(base + 1, limit);
}
```

1. (10 points) Identify the base case(s) of function puzzle(..)

2. (10 points) Identify the recursive case(s) of function puzzle(..)

3. (10 points) Show what would be displayed by the following calls.

    a. System.out.print(puzzle(14,10));

    b. System.out.print(puzzle(4,7));

    c. System.out.print(puzzle(0,0));

5.4. Problem 3: Write a recursive function that computes the sum of all numbers from 1 to n, where n is given as a parameter.

```
//return the sum 1+ 2+ 3+ ...+ n
int sum(int n)
```

5.5. Problem 5: Write a recursive function that computes and returns the sum of all elements in an array, where the array and its size are given as parameters.

```
//return the sum of all elements in a[]
int findsum(int a[], int n)
```

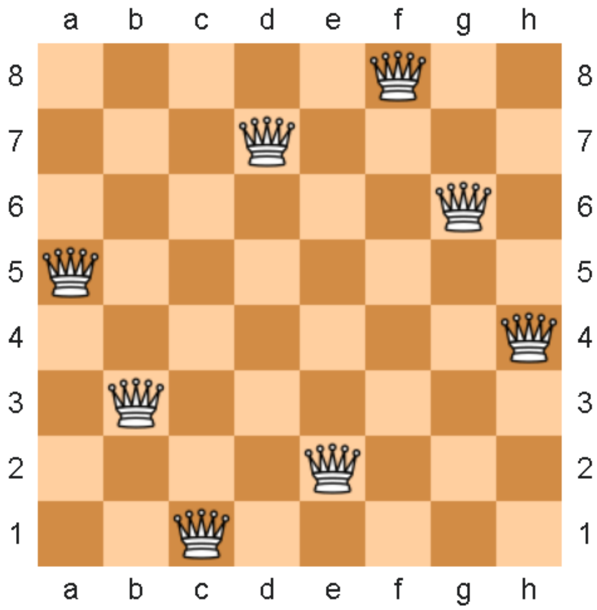
5.6. Problem 7: Write a recursive function that takes a string as input and reverses it using recursion.

5.7. Problem 8: Write a recursive function to generate all subsets of a given set.

5.8. Problem 9: Implement a recursive solution to the N-Queens Puzzle

https://en.wikipedia.org/wiki/Eight_queens_puzzle) for a given board size

- *Hint: start with N = 4 first*


a b c d e f g h
8 7 6 5 4 3 2 1