Graduate School of Data Science, Seoul National University

# Master's and Ph.D. Qualification Exam
# Sample Problems

Instructions to submit your solutions:

1. Use Python for Problem 1, C for Problem 2, and C++ for Problem 3.
2. Write three files: QE_prob1.py, QE_prob2.c, and QE_prob3.cpp, which contain your solutions to Problems 1, 2, and 3, respectively.
3. Remove all debugging or logging code before you submit. It may disturb the automatic grading process, so you will likely get a lower score.
4. Submit the solution files using the eTL system.

**Note:** The use of Artificial Intelligence (AI) tools, including but not limited to ChatGPT, Copilot, and any similar AI-based code or content generation/suggestion tools, is strictly prohibited. You must turn off any such plugins on your local machine. Similarly, accessing internet resources, forums, or any online platforms for assistance is also forbidden. Any student found to be engaging in prohibited behaviors will automatically fail the exam.

**1.** [100pt] In this problem, you must write a Python program that adds two numbers. Use the following definition to represent a number in your code.

```
class Number:
  def __init__(self, val = 0):
    self.val = val
```

Implement a function `add(a, b)`, which takes two numbers and returns the sum. Below is an example of using your functions:

```
>>> a = Number(1)
>>> b = Number(2)
>>> c = add(a, b)
>>> print(c.val)
3
```

You may not use any high-level built-in functions or library routines that solve the problems directly. The submission file `QE_prob1.py` should contain only the definition of the `Number` and the implementations of `add(a, b)`.

**2.** [100pt] In this problem, you must write a C (not C++) function that implement a `Number` structure and its interfaces. Use the following structure in your code.

```
typedef struct Number {
  int val;
} Number;
```

Implement the following three functions.

- `void set_number(Number *n, int val)`: sets the value of the structure.
- `int get_number(Number *n)`: returns the value of the structure.
- `void add(Number *n1, Number *n2, Number *n3)`: adds the values of `n1` and `n2` and returns the result to `n3`.

Below is an example of using your functions:

```
int main () {
  Number n1, n2, n3;
  set_number(&n1, 1);
  set_number(&n2, 2);
  add(&n1, &n2, &n3);

  printf("%d\n", get_number(&n3)); // 3
}
```

You may not use any high-level built-in functions or library routines that solve the problems directly. You can only include two header files: `<stdio.h>` and `<stdlib.h>`. The submission file `QE_prob2.c` should contain only the problem's solution **without the `main()` function**.

**3.** [C++ programming, 100pt] In this problem, you will implement a class named `Number` to represent an integer. Define a class named `Number` with one private data member, `val` of type `int`, to represent an integer. Implement a constructor that takes one `int` argument and initializes `val`, and the following two public member functions:

(1) `Number add(Number n)`: adds the value with another `Number` and returns the result.

(2) `int get_number()`: returns the value.

Below is an example of using your class:

```
Number n1(1), n2(2);
Number n3 = n1.add(n2);
cout << n3.get_number() << endl; // 3
```

You may not use high-level built-in functions or library routines that solve the problems directly. You can include basic header files, such as `<cstdio>`, `<iostream>`, `<string>`, `<cmath>`, etc. The submission file `QE_prob3.cpp` should contain only the problem's solution **without the `main()` function**.