# Object Oriented Programming (IGS2130)

## Lab 2

**Instructor:**
**Choonwoo Ryu, Ph.D.**


INHA UNIVERSITY

# Exercise #1

Hint
01. p19, p20

- Write a program that tells you your grade when you input your test score.
  - Exit the program when a negative score entered
  - Expected output
    - Above 90: *Congratulations. Your grade: A*
    - 89~80: *Your grade: B*
    - 79~70: *Your grade: C*
    - 69~60: *Your grade: D*
    - Below 60: *Sorry. Your grade: F*

```
Enter your score of the exam: 59
Sorry. Your grade: F

Enter your score of the exam: -1
End of the program
```

```
Enter your score of the exam: 78
Your grade: C

Enter your score of the exam: 99
Congratulations. Your grade: A

Enter your score of the exam: 90
Congratulations. Your grade: A

Enter your score of the exam: 89
Your grade: B

Enter your score of the exam: 60
Your grade: D
```

# Exercise #2

Hint
01. p29

- Create multiple display() functions with function overloading so that the following program can produce the given output.

```
char type: A
int type: 1234
double type: 123.4
```

```
#include <iostream>
using namespace std;

// function prototypes

int main() {
    display('A');
    display(1234);
    display(123.4);

    return 0;
}

// function definitions
```

# Exercise #3

Hint
01. p30

- Write cout_int() function using default parameter so that the following program can produce the given output.

```
int main() {
    cout_int(100);
    cout_int(100, 8);
    cout_int(100, 10);
    cout_int(100, 16);
    cout_int(100, 20);

    return 0;
}
```

```
100
144
100
64
ERROR: unknown base
```

Hint.

```
#include <iostream>

int main() {
    std::cout << "Decimal " << std::dec
<< 100 << '\n';
    std::cout << "Octal " << std::oct
<< 100 << '\n';
    std::cout << "Hexadecimal "
<< std::hex << 100 << '\n';

    return 0;
}
```

```
Decimal 100
Octal 144
Hexadecimal 64
```

# Exercise #4

Hint
01. p60

- Write a program that can calculate minimum, maximum, and average values from the given input data.
  - User will decide how many data will be entered
  - Use dynamic memory allocation using malloc() or new[]
  - Use pointer indexing notation
  - Create functions with pointer parameters

```
int GetMaximum(int*, int);
int GetMinimum(int*, int);
double GetAverage(int*, int);
```

  - Use them in your main function

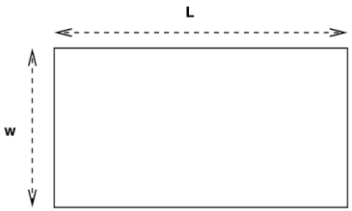
```
# of data want to input: 10
0. Input an integer: 100
1. Input an integer: -23
2. Input an integer: 1
3. Input an integer: 36
4. Input an integer: 123
5. Input an integer: 8
6. Input an integer: -12
7. Input an integer: 43
8. Input an integer: 98
9. Input an integer: 0

Minimum value is -23
Maximum value is 123
Average value is 37.4
```

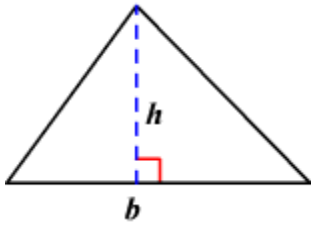
# Exercise #5

Hint
01. p23~p25, p70~p72

- Write functions that calculate rectangle, triangle, and circle area respectively.
  - Define user defined namespace named shape
  - Functions return 0 if any invalid input arguments are given
  - The given main function should run without modification
  - Area of circle: $A = \pi r^2$
  - Area of triangle: $A = 1/2bh$
  - Area of rectangle: $A = WL$

```
#define _USE_MATH_DEFINES
#include <cmath>
…
M_PI
```





```
int main() {
    cout << "Area of Rectangle(W:20, L:10):";
    cout << shape::rec_area(20, 10) << endl;

    cout << "Area of Triangle(b:20, h:10):";
    cout << shape::tri_area(20, 10) << endl;

    cout << "Area of Circle(r:20):";
    cout << shape::cir_area(20) << endl;

    return 0;
}
```

```
Area of Rectangle(W:20, L:10):200
Area of Triangle(b:20, h:10):100
Area of Circle(r:20):1256.64
```