

Lab 1 ตัวแปรและการจองพื้นที่ในหน่วยความจำ

(Variables and Memory allocation)

วัตถุประสงค์

1. สามารถติดตั้งและใช้เครื่องมือ IDE ในการพัฒนาโค้ดภาษา C++ ได้
2. เข้าใจการประกาศตัวแปรในแบบต่าง ๆ และการจองพื้นที่ในหน่วยความจำ Static, Stack และ Heap ของคอมพิวเตอร์
3. เข้าใจการทำงานกับ Pointers อย่างง่าย
4. สามารถระบุส่วน Code ที่อาจจะเกิดข้อผิดพลาดในแบบต่าง ๆ ที่เกิดจากการจองและคืนพื้นที่ในหน่วยความจำ
5. เข้าใจขอบเขต (Scope) และ ชั่วชีวิต (Lifetime) ของตัวแปร

การส่งงาน

ให้พิมพ์คำตอบลงในฟอร์มที่กำหนดให้ บนเว็บไซต์รายวิชา

การส่งงานล่าช้าจะทำให้ไม่ได้รับคะแนนในส่วนนี้

ตัวแปร (Variable) คือชื่อที่ใช้อ้างถึงหน่วยความจำ (Memory) ของคอมพิวเตอร์ที่ใช้เก็บค่าข้อมูลที่มีชนิดข้อมูลที่แตกต่างกัน เช่น จำนวนเต็ม จำนวนจริง เป็นต้น โดยที่ค่าของตัวแปรสามารถเปลี่ยนแปลงได้ตลอดเวลา ชื่อของตัวแปรอาจจะมองว่ามันคือป้ายที่บอกว่าพื้นที่ที่กำหนดไว้อยู่ที่ Address ไหนบนหน่วยความจำ อย่างไรก็ตามหน่วยความจำอาจจะไม่ได้ถูกจองตลอดไปในหนึ่งโปรแกรม กล่าวคือ ภาษาโปรแกรมนั้นมีวิธีการจองหน่วยความจำสำหรับตัวแปรอย่างชาญฉลาด โดยเราจะใช้คำว่า **“Allocate”** เพื่ออ้างถึงการจองพื้นที่บนหน่วยความจำสำหรับตัวแปรเพื่อเก็บค่าและเรา **“Deallocate”** เมื่อระบบขอคืนพื้นที่บนหน่วยความจำจากตัวแปร ดังนั้นตัวแปรดังกล่าวจะไม่สามารถใช้พื้นที่นั้นเก็บค่าได้อีกต่อไป นอกจากนั้นแล้ว **Lifetime** ของตัวแปรใช้กล่าวถึงช่วงเวลาตั้งแต่ Allocation จนถึง Deallocation และ Error ที่พบบ่อยคือการใช้พื้นที่บนหน่วยความจำที่ถูก Deallocated ไปแล้ว เพื่อให้เข้าใจแนวคิดการจองพื้นที่ในหน่วยความจำให้นักศึกษารัน Code ที่กำหนดให้และตอบคำถามในแต่ละข้อต่อไปนี้อย่างละเอียดในรูปแบบฟอร์มที่กำหนดให้

ข้อที่ 1

```
#include <iostream>
using namespace std;
int main(int argc, char **argv){
    int *x, y;

    *x = 10;
    y = 10;

    cout << *x << ", " << y << endl;
    cout << x << ", " << &y << endl;
    return 0;
}
```

- 1.1 หลังจากรันโปรแกรมบนหน้า Console ของนักศึกษาจะมีข้อความใดปรากฏขึ้นมา (หากเกิด Error ให้รายงานว่าเกิด Error messages ว่าอย่างไรบ้าง)
- 1.2 ให้นักศึกษาอธิบายว่าทำไมผลการรันจึงได้ดังนี้

ข้อที่ 2

```
#include <iostream>
```

```

using namespace std;

int squareArea(int w){
    int result;

    static int callCount = 0;

    callCount++;

    cout << "Call count: " << callCount << endl;

    result = w*w;

    return result;
}

int main(int argc, char **argv){
    cout << squareArea(5) << endl;

    cout << squareArea(6) << endl;

    cout << squareArea(7) << endl;

    return 0;
}

```

2.1 หลังจากรันโปรแกรมบนหน้า Console ของนักศึกษาจะมีข้อความใดปรากฏขึ้นมา (หากเกิด Error ให้รายงานว่าเกิด Error messages ว่าอย่างไรบ้าง)

2.2 ให้นักศึกษาลองทายว่าตัวแปรใดบ้างถือเป็น Local (Stack) variables สำหรับฟังก์ชัน squareArea(int w)

ข้อ 3

```

#include <iostream>

using namespace std;

int* rectangleArea(int w, int h){
    int* result = (int*) malloc (sizeof(int));

    *result = w*h;

    w++; h++;

    return result;
}

int main(int argc, char **argv){
    int a = 7, b = 3;

    int* x = rectangleArea(a, b);

    cout << x << " " << *x << endl;

    cout << a << " " << b << endl;

    return 0;
}

```

}

3.1 หลังจากรันโปรแกรมบนหน้า Console ของนักศึกษาจะมีข้อความใดปรากฏขึ้นมา (หากเกิด Error ให้รายงานว่าเกิด Error messages ว่าอย่างไรบ้าง)

3.2 นักศึกษาคิดว่า (หรือเดาว่า) ในโปรแกรมด้านบนนี้ ณ ขณะทำงาน มี Pointers มากสุดกี่ตัวเกิดขึ้นใน Program Stacks (ทั้งของฟังก์ชัน `main()` และ `rectangleArea(•)`)

ข้อ 4

```
#include <iostream>

using namespace std;

int* rectangleArea2(int* w, int& h){

    int* result = (int*) malloc (sizeof(int));

    *result = *w*h;

    w++; h++;

    return result;

}


int main(int argc, char **argv){

    int a = 7, b = 3;

    int* x = rectangleArea2(&a, b);

    cout << x << " " << *x << endl;

    cout << a << " " << b << endl;

    return 0;

}
```

4.1 หลังจากรันโปรแกรมบนหน้า Console ของนักศึกษาจะมีข้อความใดปรากฏขึ้นมา (หากเกิด Error ให้รายงานว่าเกิด Error messages ว่าอย่างไรบ้าง)

4.2 ให้อธิบายว่าทำไมผลลัพธ์ของค่า a และ b จึงเป็นดังนั้น

[illegible]