

การทดลองที่ 1.1 ให้นักศึกษา พิมพ์และทำการทดลองโปรแกรมต่อไปนี้

1	#include <iostream>
2	int main(){
3	cout << “Hello World”;
4	}

บันทึกผลการทดลอง

code error เพราะว่า ตัวโปรแกรมยังไม่มีเรียกใช้ library ไม่รู้จักตัว cout

การทดลองที่ 1.2 ให้นักศึกษา พิมพ์และทำการทดลองโปรแกรมต่อไปนี้

1	#include <iostream>
2	int main(){
3	std::cout << “Hello World”;
4	}

บันทึกผลการทดลอง

โปรแกรมใช้งานได้ปกติเพราะว่ามีการเรียกใช้ std:: cout ซึ่งดึง library มาเฉพาะส่วน
cout

การทดลองที่ 1.3 ให้นักศึกษา พิมพ์และทำการทดลองโปรแกรมต่อไปนี้

1	#include <iostream>
2	using namespace std;
3	int main(){
4	cout << “Hello World”;
5	}

บันทึกผลการทดลอง

โปรแกรมใช้งานได้ปกติ เพราะมีการเรียกใช้ library ทั้งหมดของตัว std ดัง **using namespace std;**

คำถาม

จากผลการทดลองที่ 1.2 และ 1.3 มีความเหมือนหรือต่างกันอย่างไร

เหมือนกันตรงที่สามารถแสดงผลลัพธ์ออกมาได้เหมือนกันและใช้ library เดียวกันแต่แตกต่างตรงที่ 1.3 มีการเรียกใช้ library ทั้งหมดออกมาใช้โดยใช้ **using namespace std;** ประการรอบเดียวแต่ 1.2 มีการเรียกใช้ทีละรอบ **std:: cout**

จากผลการทดลองที่ 1.2 และ 1.3 จึงไม่เกิด Syntax Error และสามารถแสดงผลลัพธ์ได้

ที่แสดงผลลัพธ์ได้เพราะตัวโปรแกรมมีการ เรียกใช้ library std เหมือนกัน

การทดลองที่ 1.4 ให้นักศึกษา พิมพ์และทำการทดลองโปรแกรมต่อไปนี้

1	#include <iostream>
2	using namespace std;
3	int main(){
4	cout << "Hello World";
5	return (0);
6	}

บันทึกผลการทดลอง

โปรแกรมรันได้ปกติจะได้ผลลัพธ์ Hello World

จากผลการทดลองที่ 1.3 และ 1.4 มีความเหมือนหรือต่างกันอย่างไร

แตกต่างกันเพราะ 1.3 ไม่มีการคืนค่า แต่ 1.4 มีการคืนค่าแต่ผลลัพธ์ที่ออกมาเหมือนกัน

จากผลการทดลองที่ 1.3 และ 1.4 มีคำสั่งโปรแกรมอะไรที่ต่างกัน

1.4 มี return 1.3 ไม่มี return

การทดลองที่1.5 ให้นักศึกษาพิมพ์และทำการทดลองโปรแกรมต่อไปนี้

1	#include <iostream>
2	using namespace std;
3	int main(){
4	cout << “Welcome to ”;
5	cout << “C++ language”;
6	return (0);
7	}

บันทึกผลการทดลอง

โปรแกรมรันได้ปกติ จะได้ผลลัพธ์ Welcome to C++ language

คำถาม

ถ้าหากต้องการให้ผลลัพธ์ที่ปรากฏในหนึ่งบรรทัดจำเป็นต้องใช้ cout

เพียงคำสั่งเดียวหรือไม่

ไม่จำเป็นต้องใช้cout คำสั่งเดียวก็ได้สามารถใช้ cout

หลายคำสั่งติดกันในบรรทัดเดียวกันได้

การทดลองที่1.6 ให้นักศึกษาพิมพ์และท าการทดลองโปรแกรมต่อไปนี้

1	#include <iostream>
2	using namespace std;
3	int main(){
4	cout << “This is structure of C++ language.” << endl;
5	return (0);
6	}

บันทึกผลการทดลอง

โปรแกรมรันได้ปกติ จะได้ผลลัพธ์ This is structure of C++ language.

การทดลองที่1.6 การแสดงข้อความ This is structure of C++ language.” บนหน้าจอ

1	#include <iostream>
2	using namespace std;
3	int main(){
4	cout << “This is structure of C++ language.” << endl;
5	return (0);
6	}

บันทึกผลการทดลอง

โปรแกรมรันได้ปกติ จะได้ผลลัพธ์ This is structure of C++ language.

การทดลองที่ 1.7 การแสดงข้อความ This is structure of C++ language.” บนหน้าจอ

1	#include <iostream>
2	using namespace std;
3	int main(){
4	cout << “This is structure of ” << endl;
5	cout << “C++ language.” << endl;
6	return (0);
7	}

บันทึกผลการทดลอง

โปรแกรมรันได้ปกติ จะได้ผลลัพธ์ This is structure of (บรรทัด1)

C++ language.(บรรทัด2)

จากผลการทดลองที่ 1.6 และ 1.7 มีความเหมือนหรือต่างกันอย่างไร

1.6 มีการรันคำสั่งรวมกันใน บรรทัดเดียว แต่ 1.7 มีการรันคำสั่งแยกกัน 2 บรรทัด

จากผลการทดลองที่ 1.7 ในบรรทัดที่ 5 ถ้าไม่มีการใช้ endl ปิดท้ายคำสั่ง

ผลลัพธ์จะเป็นอย่างไร

ผลลัพธ์จะเป็นเหมือนเดิมไม่มีอะไรเปลี่ยนแปลง

คำสั่ง endl มีหน้าที่อะไร

ใช้สำหรับ ขึ้นบรรทัดใหม่

นำเสนอส่วนของ Code ที่ทำให้ได้ผลลัพธ์ในรูปแบบเดียวกัน ด้วย cout คำสั่งเดียว

```
cout << "This is structure of " << endl;
```

```
cout << "C++ language." << endl;
```

```
cout << "This is structure of " << endl << "C++ language." << endl;
```