# ใบงานการทดลองที่ 7 เรื่อง Overloading Method และ Overriding Method

## จุดประสงค์ทั่วไป

### รู้และเข้าใจการพ้องรูปในการเขียนโปรแกรมเชิงวัตถุ

### รู้และเข้าใจการสืบทอดของวัตถุ โครงข่ายของวัตถุ โครงสร้างของโปรแกรมเชิงวัตถุ

## เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

## ทฤษฎีการทดลอง

### Super Class คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

|  |
| --- |
| คือ class ที่อยู่บนสุดเป็น class แม่ของ subclass |
| Super class |
| Subclass Subclass |

### การพ้องรูป(Polymorphism) คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

|  |
| --- |
| คือการที่ class แม่แปลงร่างเป็น class ลูกเพื่อใช้ method และ property ของ class ลูกได้ |
|  |
|  |
|  |

### Overloading Method คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

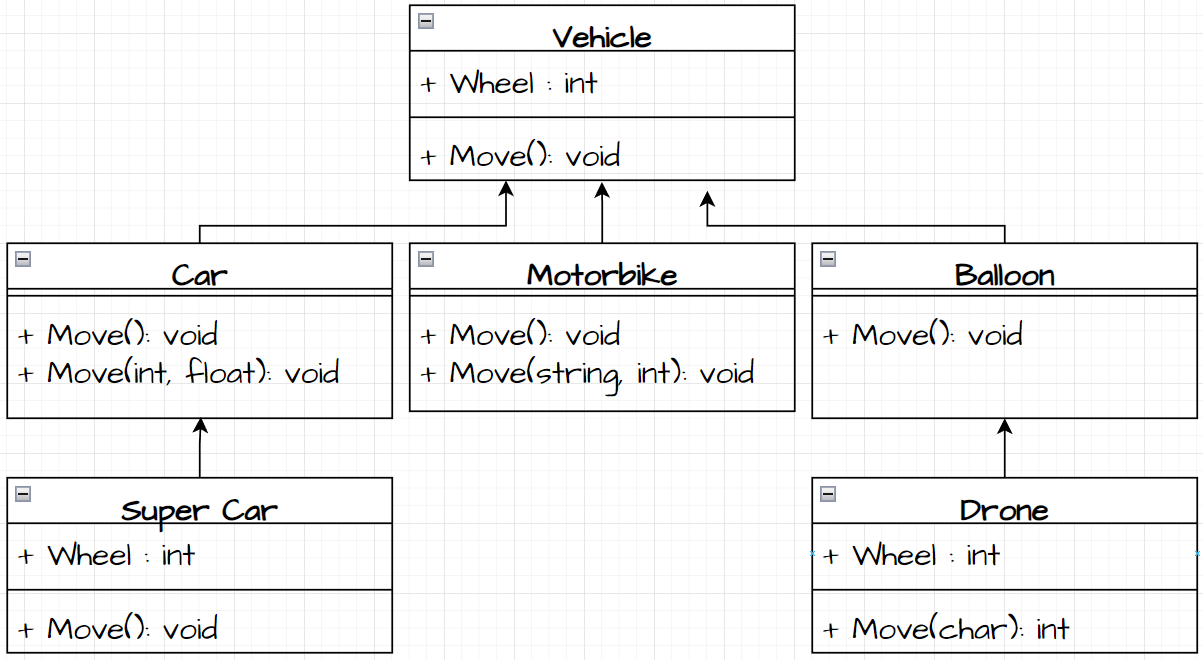
|  |
| --- |
| คือ Method ที่มีชื่อเหมือนกัน แต่มี Parameter ต่างกัน |
|  |
|  |
|  |

### Overriding Method คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

|  |
| --- |
| คือ method ที่มีชื่อเหมือนกันกับ class แม่ แต่มีรายละเอียดที่ต่างกัน |
|  |
|  |
|  |

## ลำดับขั้นการปฏิบัติการ

### จงสร้างคลาสต่างๆ และทำการสืบทอดกันตามโครงสร้างดังต่อไปนี้ โดยแต่ละ Method จะต้องสร้างด้วยชื่อเดียวกัน แต่มีกระบวนการทำงานที่แตกต่างกัน พร้อมทั้งแสดงผลลัพธ์การทำงานภายในแต่ละคลาสให้ดูเพื่อแสดงถึงความแตกต่าง ตามหลักการของ Overloading Method และ Overriding Method



### จงเขียนโค้ดโปรแกรมจากผังงานดังกล่าว ที่แสดงให้เห็นว่าเป็น Overloading Method

|  |
| --- |
| โค้ดโปรแกรม |
| **public** **class** vehicle {  **public** **int** wheel ;    **public** vehicle() {  System.***out***.println("\n | Vehicle start |");  }//end method    /\*\*  \* Start move vehicle  \*/  **public** **void** move() {  System.***out***.print(" | Vehicle Move |");  }//end method    /\*\*  \* Just moving vehicle  \* **@param** name user input name Driver  \*/    **public** **void** move( String name) {  System.***out***.println("--|My Driver ::" + name +" --->");  }//end method  } |

### จงเขียนโค้ดโปรแกรมจากผังงานดังกล่าว ที่แสดงให้เห็นว่าเป็น Overriding Method

|  |
| --- |
| โค้ดโปรแกรม |
| **package** main\_lap07;  **public** **class** car **extends** vehicle {  /\*\*  \* Start move car  \*/  **public** car() {  System.***out***.println(" | car start ! |");  }//end method    /\*\*  \* Just moving a car  \* **@param** name user input speed for car and kilometer/hour  \*/    **public** **void** move(**int** speed , **float** km) {  System.***out***.println(" My Driver , drive at speed "+ speed + " km/hr form home " + km + " kilometer");  }//end method  }//end class |

### จงเขียนโค้ดโปรแกรมแสดงการเรียกใช้งานในการสร้างวัตถุทั้งหมดเพื่อมาทดสอบในฟังก์ชันหลัก

|  |
| --- |
| โค้ดโปรแกรม |
| **package** main\_lap07;  **public** **class** main {  **public** **static** **void** main (String[] args) {  vehicle vehicle = **new** vehicle();  vehicle.move();  vehicle.move("Pawit");    car car = **new** car();  car.move(90, 50);    supercar supercar = **new** supercar();  supercar.move();    motobike motobike = **new** motobike();  motobike.move("honda", 150);    ballon balloon = **new** ballon();  balloon.move();    drone drone = **new** drone();  drone.move('a', 150);;    }  } |

### ผลลัพธ์การทำงานของโปรแกรม

|  |
| --- |
| ผลลัพธ์การทำงาน (พร้อมคำอธิบายประกอบ) |
|  |

## สรุปผลการปฏิบัติการ

|  |
| --- |
| จากการปฏิบัติพบว่าการส่งข้อมูลผ่าน class จะต้องระวังเรื่องการเรียกใช้งานถ้าประกาศผิดก็จะไม่ |
| สามารถเรียกใช้งานได้ |
|  |
|  |
|  |
|  |

## คำถามท้ายการทดลอง

### ข้อแตกต่างระหว่าง Overloading method และ Overriding method คืออะไร?

|  |
| --- |
| แตกต่างกันตรงที่ Overloading method เป็น Method ที่มีชื่อเหมือนกัน แต่มี Parameter ต่างกัน |
| ส่วน Overriding method จะเป็น method ที่มีชื่อเหมือนกันกับ class แม่ แต่มีรายละเอียดที่ต่างกัน |
|  |
|  |

### คุณคิดว่าหลักการของ Polymorphism จะสามารถนำมาประยุกต์ใช้ในการเขียนโปรแกรมของคุณได้อย่างไรบ้าง ?

|  |
| --- |
| ประยุกต์ใช้ในการที่เราต้องการสร้าง class แม่ แล้วเราต้องการเรียกใช้งาน method ของ class ลูก |
|  |
|  |