



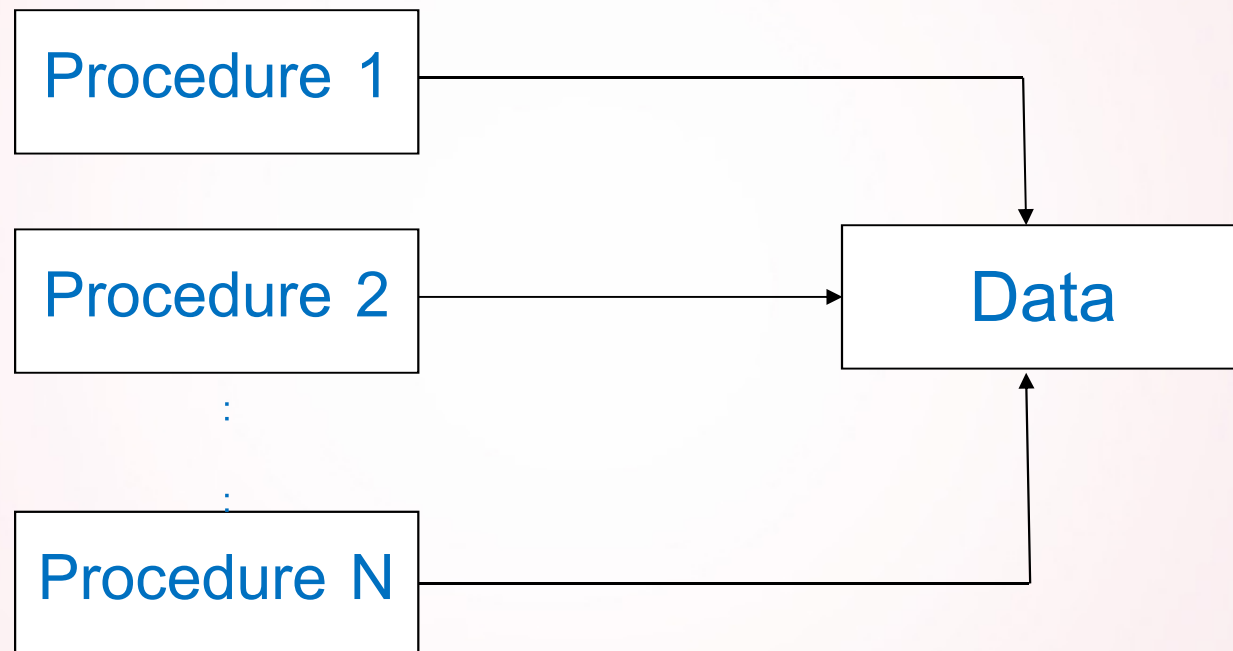
การพัฒนาโปรแกรมเชิงวัตถุ (Object-oriented program development)

**Matinee Kiewkanya
Computer Science, CMU**

การเขียนโปรแกรมแบบดั้งเดิม

- ❁ นิยมใช้การเขียนโปรแกรมในรูปแบบที่เรียกว่า การเขียนโปรแกรมเชิงกระบวนการ (Procedural Programming) ซึ่งจัดว่าเป็นการเขียนโปรแกรมแบบโครงสร้าง (Structured Programming)
- ❁ จุดประสงค์คือเพื่อให้เครื่องคอมพิวเตอร์สามารถวิเคราะห์ ข้อมูล ประมวลผล และแก้ปัญหาบางอย่างแทนมนุษย์
- ❁ หลักการคือ ให้แก้ปัญห ด้วยการแบ่งปัญหาออกเป็น ส่วน ๆ ที่เรียกว่า กระบวนการ (Procedure) หรือฟังก์ชัน (Function) หรือ โมดูล (Module)
- ❁ แก้ปัญหาในแต่ละส่วนด้วยขั้นตอนวิธี ที่เรียกว่า อัลกอริทึม (Algorithm) และมีการใช้งานโครงสร้างข้อมูล (Data Structure) เฉพาะ เพื่อจัดเก็บข้อมูล

การเขียนโปรแกรมแบบดั้งเดิม



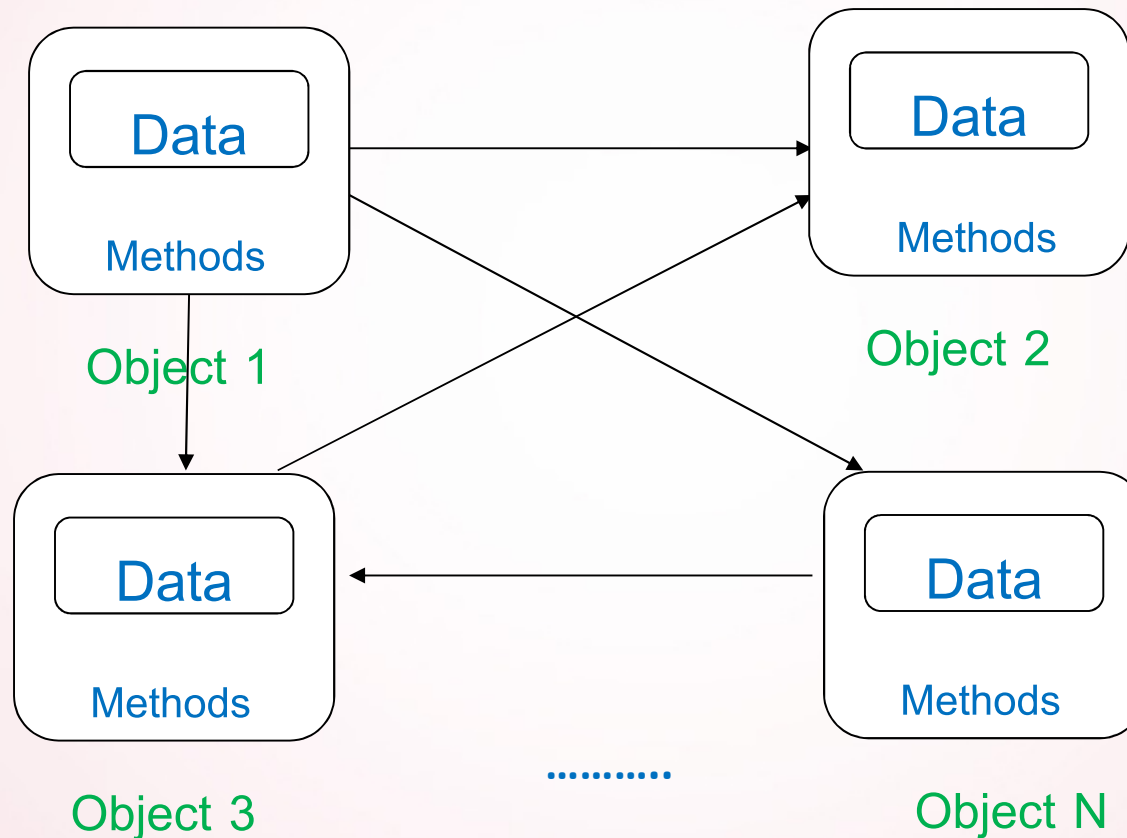
ข้อเสียของการเขียนโปรแกรม แบบดั้งเดิม

- ❁ ไม่สามารถควบคุมการไหลของข้อมูลและไม่สามารถปกป้องข้อมูลได้
- ❁ ขาดลักษณะของการนำโปรแกรมกลับมาใช้ใหม่ในภายหลัง
- ❁ ในกรณีที่ปัญหานั้นมีขนาดใหญ่ จะต้องแบ่งระบบออกเป็นส่วนย่อย ๆ เป็นจำนวนมาก ทำให้ขนาดของโปรแกรมที่จะพัฒนาเพื่อแก้ปัญหานั้นมีขนาดใหญ่ตาม และมีความซับซ้อนของรหัสต้นฉบับ (Source Code) มากขึ้น

การเขียนโปรแกรมเชิงวัตถุ

- ❁ หลักการของการพัฒนาโปรแกรมเชิงวัตถุ จะมีมุมมองต่อปัญหาเป็นวัตถุ
- ❁ ทำการแยกปัญหาหรือระบบออกเป็นวัตถุ โดยที่แต่ละวัตถุจะรวมเอาส่วนของข้อมูลและกรรมวิธีในการจัดการข้อมูลเข้าไว้ด้วยกันแล้วนำวัตถุต่าง ๆ มาเรียงต่อกัน
- ❁ ผู้ใช้ไม่ต้องเข้าใจลักษณะภายในของวัตถุนั้นๆ เพียงแต่ทราบว่า วิธีการที่จะเรียกใช้วัตถุแต่ละชิ้นเป็นอย่างไรก็เพียงพอแล้ว

การเขียนโปรแกรมเชิงวัตถุ



วัตถุ

❁ **วัตถุ (Object)** ในที่นี้คือกลุ่มก้อนของสิ่งที่สนใจ (Entity) ที่ประกอบไปด้วย

- **คุณสมบัติหรือคุณลักษณะ (Attribute/Property)** ซึ่งจะอธิบายได้ว่าวัตถุชนิดนี้มีลักษณะอย่างไร
- **พฤติกรรม (Behavior/Method/Function/Operation)** ซึ่งจะอธิบายถึงการกระทำที่วัตถุนั้นสามารถกระทำได้

❁ ตัวอย่างของวัตถุที่พบเห็นได้บ่อย เช่น รถยนต์ ซึ่งจะมี **คุณสมบัติ**คือ ประเภทรถ ยี่ห้อ สี ป้ายทะเบียน เป็นต้น และมี **พฤติกรรม** เช่น เบรก เลี้ยวซ้าย เลี้ยวขวา ปิดน้ำฝน เปิดไฟ เป็นต้น

อะไรคือ attribute และ function
ของโทรศัพท์มือถือ ?



วัตถุ

- ❁ คำว่าวัตถุในความหมายของการพัฒนาโปรแกรมเชิงวัตถุนั้น หมายถึงรวมถึงสิ่งทั้งที่เป็นรูปธรรม หรือนามธรรม ซึ่งเป็นสิ่งใดก็ตามที่สามารถแสดงลักษณะใด ๆ ก็ได้ ซึ่งลักษณะเหล่านี้จะเป็นคุณสมบัติเฉพาะของแต่ละวัตถุที่สามารถแปรเปลี่ยนได้ตลอดการคงอยู่ของวัตถุนั้น

รีโมต



วิทยุ

คอมพิวเตอร์



คน

กฎหมาย

รัฐธรรมนูญ

- ❁ การแปรเปลี่ยนสถานะของวัตถุจะเกิดขึ้นเมื่อมีการส่งข่าวสาร (Message) ไปยังวัตถุที่ต้องการ เมื่อวัตถุดังกล่าวได้รับข่าวสาร ก็จะแสดงพฤติกรรมออกมาตอบสนองต่อข่าวสารที่ได้รับ วัตถุจึงสามารถนำมาประกอบกันและนำมาทำงานร่วมกันได้ โดยการแลกเปลี่ยนข่าวสารเพื่อนำมาประมวลผลและส่งข่าวสารที่ได้ไปให้ วัตถุ อื่นๆที่เกี่ยวข้องเพื่อให้ทำงานต่อไปได้

ตัวอย่างภาษาโปรแกรม

- ❁ ตัวอย่างภาษาโปรแกรม (Programming Language) ที่สนับสนุนการพัฒนาโปรแกรมเชิงวัตถุ เช่น Ada C++ C# Java Javascript Perl PHP Visual Basic Python และ Ruby เป็นต้น
- ❁ ภาษาเหล่านี้จะสนับสนุนคุณสมบัติสำคัญในการเขียนโปรแกรมเชิงวัตถุ

ลักษณะที่สำคัญของภาษาที่ใช้ในการพัฒนาโปรแกรมเชิงวัตถุ

1. การห่อหุ้ม (Encapsulation) แปลจากคำศัพท์จะหมายถึงการรวมกันโดยมีเปลือกห่อหุ้ม ซึ่งในการพัฒนาโปรแกรมเชิงวัตถุ จะหมายถึงการรวมกันของโครงสร้างข้อมูลกับฟังก์ชันที่ใช้จัดการข้อมูลเหล่านั้น หรือกล่าวอีกอย่างหนึ่งคือการห่อหุ้มนี้จะหมายถึงการรวมกันของแอตทริบิวต์และเมทอดของวัตถุนั้นเอง
 - ผลดีคือการใช้งานผ่านเมทอดจะมีความยืดหยุ่นสูงกว่าการใช้งานแอตทริบิวต์ของวัตถุโดยตรง แต่ละวัตถุมีความเป็นอิสระจากกัน
 - การดูแลรักษาโปรแกรม (Maintain) ทำได้ง่าย เพราะหากมีการเปลี่ยนแปลงภายในวัตถุใด ๆ แต่ถ้าส่วนติดต่อกับวัตถุอื่นยังคงเดิม ก็จะไม่ส่งผลกระทบกับทั้งระบบ

ลักษณะที่สำคัญของภาษาที่ใช้ในการพัฒนาโปรแกรมเชิงวัตถุ

- นอกจากนั้นการซ่อนข้อมูลจะทำให้วัตถุสามารถ**ซ่อนสารสนเทศ (Information Hiding)** จากภายนอก การเข้าถึงข้อมูลจะสามารถทำได้โดยผ่านส่วนต่อประสาน โดยผู้ใช้ไม่จำเป็นต้องทราบรายละเอียดภายใน



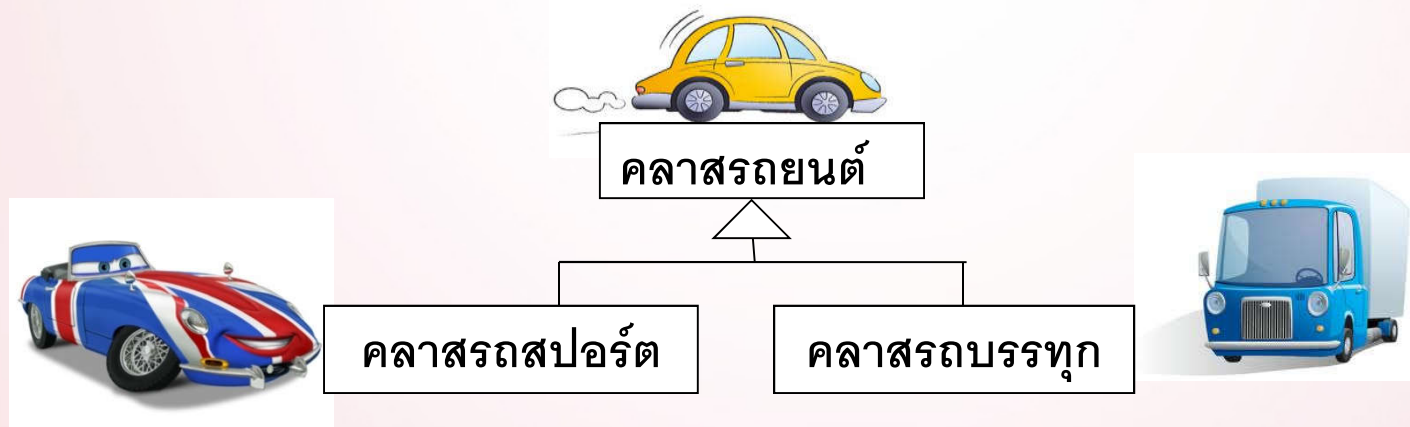
การใช้งานโทรทัศน์ที่มีการซ่อนฟังก์ชันการทำงานไว้ภายใน

ลักษณะที่สำคัญของภาษาที่ใช้ในการพัฒนาโปรแกรมเชิงวัตถุ

2. การสืบทอดคุณสมบัติ (Inheritance) จะเป็นการสร้างคลาสใหม่ขึ้นมาโดยมีพื้นฐานมาจากคลาสเดิม
- ในการเขียนโปรแกรมเชิงวัตถุ นั้น วัตถุในชีวิตจริงจะถูกระบุให้อยู่ในรูปแบบของคลาส (Class) ซึ่งจะประกอบด้วยแอตทริบิวต์และเมทอดของคลาส โดยคลาสจะเป็นต้นแบบของวัตถุ
 - ในการสืบทอดคุณสมบัติ คลาสใหม่จะมีลักษณะเดียวกันกับคลาสพื้นฐานแต่จะมีลักษณะพิเศษที่เพิ่มเติมขึ้นมามีด้วย ซึ่งจะมีความเฉพาะเจาะจงมากยิ่งขึ้น

ลักษณะที่สำคัญของภาษาที่ใช้ในการพัฒนาโปรแกรมเชิงวัตถุ

- คลาสที่เป็นคลาสพื้นฐานหรือคลาสแม่จะเรียกว่า **Base class** และคลาสที่สืบทอดมาจากคลาสพื้นฐานหรือคลาสลูกจะเรียกว่า **Derived class**
- ข้อดีของการสืบทอดคือจะช่วยให้ไม่ต้องพัฒนารหัสซ้ำกันในส่วนที่ซ้ำกันหลาย ๆ รอบ



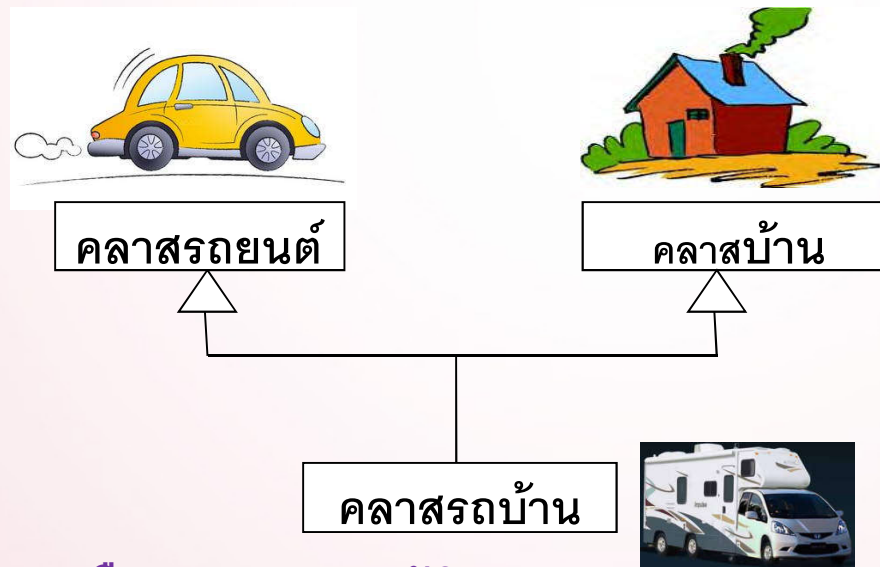
การสืบทอดคุณสมบัติจากคลาสรถยนต์สู่คลาสรถสปอร์ตและคลาสรถบรรทุก

ลักษณะที่สำคัญของภาษาที่ใช้ในการพัฒนาโปรแกรมเชิงวัตถุ

- **คลาสรายนต์** จะมีคุณลักษณะคือ เป็นยานพาหนะทางบกที่ขับเคลื่อนที่ด้วยพลังงานอย่างใดอย่างหนึ่งและถ่ายทอดลงสู่ล้อ เพื่อพาผู้ขับ ผู้โดยสาร หรือสิ่งของไปยังจุดหมายปลายทาง
- **คลาสบรรทุก**ซึ่งสืบทอดมาจากคลาสรายนต์ จะมีลักษณะพื้นฐานเช่นเดียวกับรายนต์ แต่จะมีความสามารถที่เพิ่มมาคือการบรรทุกของ
- ส่วน**คลาสสปอร์ต**ที่สืบทอดมาจากคลาสรายนต์ เช่นเดียวกัน แต่จะมีความสามารถที่เพิ่มขึ้นคือสมรรถนะในการขับเคลื่อนที่รวดเร็วกว่ารายนต์ทั่วไป

ลักษณะที่สำคัญของภาษาที่ใช้ในการพัฒนาโปรแกรมเชิงวัตถุ

- นอกจากนั้นคลาสหนึ่ง ๆ ยังสามารถสืบทอดคุณสมบัติจากคลาสพื้นฐานได้มากกว่าหนึ่งคลาส เรียกว่า **การสืบทอดคุณสมบัติแบบหลายทาง (Multiple Inheritance)**



การสืบทอดคุณสมบัติแบบหลายทางจากคลาสรถยนต์
และคลาสบ้านสู่คลาสรถบ้าน

ผศ.ดร.เมทินี เชี่ยวกันยະ ภาควิชาวิทยาการคอมพิวเตอร์ มช.

ลักษณะที่สำคัญของภาษาที่ใช้ในการพัฒนาโปรแกรมเชิงวัตถุ

3. พหุลักษณ์ หรือโพลิมอร์ฟิซึม (Polymorphism)

- เป็นภาษากรีก แปลว่ามีหลายรูปแบบ (Having many shapes) ในการพัฒนาโปรแกรมเชิงวัตถุ พหุลักษณ์คือ ภาวะที่เมทอดมีการทำงานได้หลายรูปแบบ
- ถึงแม้ว่าจะมีการเรียกใช้เมทอดเดียวกัน แต่ผลลัพธ์จากการทำงานของเมทอดจะแตกต่างกันออกไป



คลาสม้า



คลาสนก



คลาสสุนัข

พหุลักษณ์ของการแปลงเสียงในคลาสม้า คลาสนก และคลาสสุนัข

ลักษณะที่สำคัญของภาษาที่ใช้ในการพัฒนาโปรแกรมเชิงวัตถุ

- สมมติว่าคลาสแม่คือคลาสสัตว์ ซึ่งสืบทอดเมทอดการแปลงเสียง ให้แก่คลาสลูกทั้งสาม คือ คลาสม้า คลาสนก และคลาสสุนัข
- คลาสสุนัข โดยทั้งคลาสม้า คลาสนก และคลาสสุนัข นั้นมีเมทอดของการแปลงเสียงเดียวกัน
- เมื่อได้รับข่าวสารที่บอกให้ทำการแปลงเสียง ทั้งม้า นก และสุนัขจะสามารถแปลงเสียงได้ แต่จะมีเสียงที่แตกต่างกัน เป็นต้น

ข้อดีของการพัฒนาโปรแกรมเชิงวัตถุ

1.ทำให้นักพัฒนาสามารถจำลอง (Model) ระบบได้อย่างครบถ้วน

- **ในระบบที่พัฒนาด้วยวิธีเชิงวัตถุนั้น ผู้พัฒนาจะใช้แบบจำลองเดียวกันตลอดการพัฒนา การกำหนดคลาสและวัตถุในแบบจำลองที่ได้จากการวิเคราะห์และออกแบบระบบจะสามารถเปลี่ยนเป็นรหัสต้นฉบับได้โดยตรง ดังนั้นผู้พัฒนาจึงสามารถดูความสัมพันธ์ระหว่างการกำหนดปัญหาและวิธีแก้ปัญหาดังง่าย**

ข้อดีของการพัฒนาโปรแกรมเชิงวัตถุ

2. ช่วยเพิ่มความเข้าใจในแบบจำลองของปัญหา

- การจำลองระบบของการพัฒนาเชิงวัตถุนั้นจะแสดงความสัมพันธ์ระหว่างฟังก์ชันกับข้อมูลอย่างใกล้ชิด เนื่องจากจำลองมาจากเหตุการณ์ในชีวิตจริง
- วัตถุทุกชั้นที่เกี่ยวข้องกับปัญหาจะต้องมีการนิยามลักษณะของวัตถุ โดยการให้แนวคิด (Concept) กับวัตถุนั้น ๆ ผ่านกระบวนการการแยกแยะเอกลักษณ์ (Abstraction) ตลอดจนการตั้งชื่อวัตถุก็สอดคล้องกับชื่อที่มีอยู่จริงในโดเมนของปัญหา

ข้อดีของการพัฒนาโปรแกรมเชิงวัตถุ

3. ช่วยเพิ่มเสถียรภาพของการเปลี่ยนแปลง

- การเปลี่ยนแปลงความต้องการในระบบสามารถทำได้ โดยการเพิ่มหรือเปลี่ยนแปลงภายในวัตถุหนึ่ง ๆ เท่านั้น โดยไม่ต้องเปลี่ยนแปลงโครงสร้างของระบบทั้งหมด
- นอกจากนั้นการซ่อนรายละเอียด (Encapsulation) และพหุลักษณะ (Polymorphism) ยังช่วยลดผลกระทบที่จะเกิดขึ้นเมื่อมีการแก้ไขโปรแกรม ซึ่งจะทำให้ระบบนั้นมีความง่ายในการบำรุงรักษา

ข้อดีของการพัฒนาโปรแกรมเชิงวัตถุ

4. สนับสนุนการนำกลับมาใช้ใหม่

- การพัฒนาเชิงวัตถุนั้นจะมีคุณสมบัติที่สนับสนุนการนำรหัสต้นฉบับกลับมาใช้ใหม่โดยตรงด้วยหลักการของ **Generalization** และ **Refinement**
- สำหรับ **Generalization** หรือการสืบทอดคุณสมบัติ นั้นจะสนับสนุนการนำกลับมาใช้ใหม่โดยให้ผู้ที่จะใช้ทำการเพิ่มคุณสมบัติของคอมโพเนนท์ (**Component**) โดยที่ไม่ต้องมีการเปลี่ยนแปลงรหัสต้นฉบับเดิม เพียงแต่เพิ่มรหัสต้นฉบับในส่วนที่แตกต่างจากที่มีอยู่ในคอมโพเนนท์เดิมเท่านั้น
- ส่วน **Refinement** จะคล้ายกับ **Generalization** แต่จะยอมให้นักพัฒนาใช้วัตถุที่ไม่สมบูรณ์โดยการเพิ่มส่วนที่หายไป

ข้อดีของการพัฒนาโปรแกรมเชิงวัตถุ

5. สนับสนุนการปรับเปลี่ยนขนาดของระบบ

- ระบบที่พัฒนาด้วยวิธีเชิงวัตถุนั้นมีการแยกแยะเอกลักษณ์และการซ่อนรายละเอียด ส่งผลให้คอมโพเนนท์แต่ละส่วนแยกออกจากกันอย่างชัดเจน และมีการใช้เครื่องหมาย (Notation) เดียวกันตลอดกระบวนการของการพัฒนา ทำให้นักพัฒนาสามารถเปลี่ยนจากการวิเคราะห์เป็นการออกแบบได้ง่าย

ข้อดีของการพัฒนาโปรแกรมเชิงวัตถุ

6. สนับสนุนการออกแบบระบบที่เชื่อถือได้และมีความปลอดภัย

- ระบบมีความน่าเชื่อถือมากกว่าเพราะผู้พัฒนาสามารถควบคุมการติดต่อระหว่างคอมโพเนนต์ได้
- ภาษาในการเขียนโปรแกรมเชิงวัตถุยังรองรับการตรวจสอบข้อบกพร่อง (Exception) เพื่อให้ผู้พัฒนาสามารถตรวจสอบและจัดการกับข้อผิดพลาดที่อาจจะเกิดขึ้นได้
- การพัฒนาเชิงวัตถุสนับสนุนการนำกลับไปใช้ใหม่ซึ่งคอมโพเนนต์ที่ได้รับการทดสอบว่าถูกต้องแล้วจะสามารถนำไปใช้กับระบบใหม่ได้โดยที่ไม่ก่อให้เกิดปัญหา