CS65: Introduction to Computer Science

Classes and Objects

Md Alimoor Reza
Assistant Professor of Computer Science

Python Data Types

- **Primitive data types:**
  - integer, floating point, boolean, etc

```
cost = 100

score = 10.56

is_present_in_class = True
```

- **Complex data types:**
  - Sequence (String, List, Tuple)
  - Dictionary

```
cost_list = [100, 200, 300]

last_name = "Reza"
```

- **Custom data type:**
  - Class

```
student = ???

person = ???

exam = ???

pet = ???
```

Class: Custom Data Type

- **Class is a Custom data type:**
  - We can create our own new types with a class definition
  - Class definition is a 'blueprint' of what should be included in a value of this type

- **Same operations can be done on this custom data type:**
  - Can create a variable of this new type
  - Change the value of the variable of this new type
  - Create a List with variables of this new type

Topics

- What is a ‘Class’?
- What is an ‘Object’?
- Difference/connection between Class and Object

- Class components
  - Initializer/Constructor
  - Attributes
  - Methods

Object Oriented Programming (OOP)

- Focus so far (up to previous lecture) has been on basic algorithmic programming structures

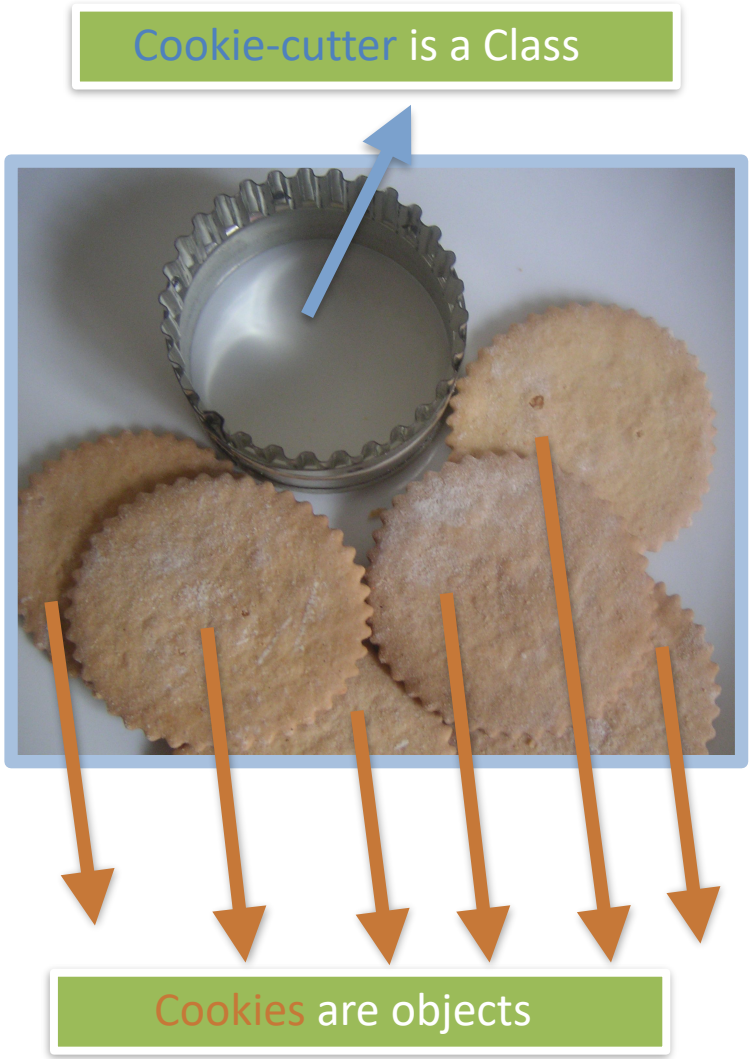
- Now, we will turn our focus on how data is stored and manipulated
  - Object Oriented Programming
  - We will still be developing algorithms

- It's a different way of thinking about writing code

Object Oriented Programming (OOP)

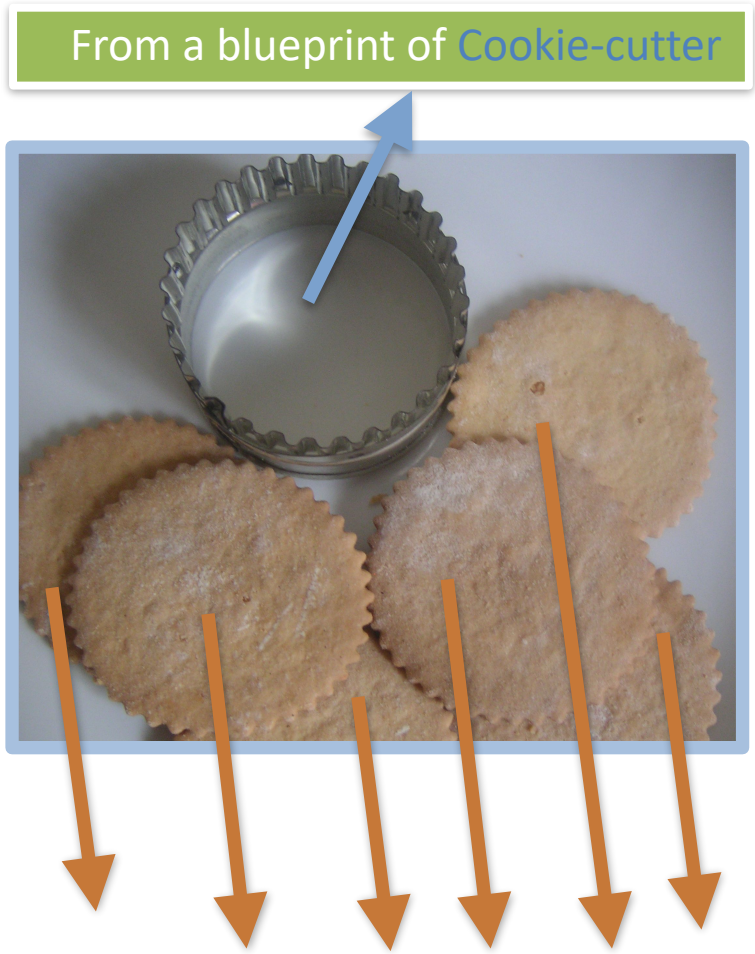
- **Class** is the blueprint or template for creating an object
- **Object:** a combination of data and associated procedures created based on the blueprint of class
- **Object Oriented Programming (OOP)** is centered on creating Objects

Object Oriented Programming (OOP)



Cookies are being curved out based on the cookie-cutter's shape

Object Oriented Programming (OOP)

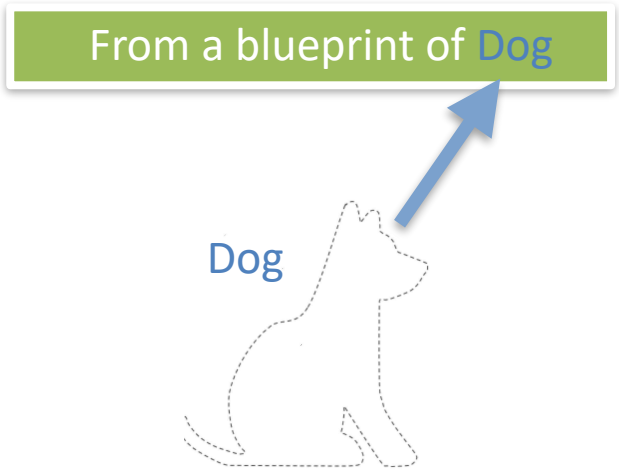


- **Class** is a blueprint or template that defines what attribute and methods **Objects** can have

- **Cookies** are made with a **Cookie-cutter** — **Objects** are made from a **Class**

- **Class** is a shape with which many individual Objects can be created — in the same way Cookie-cutter is a shape with which many cookies can be created

Object Oriented Programming (OOP)

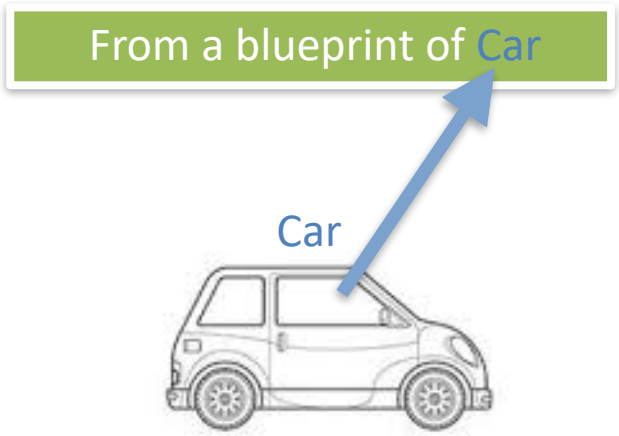


- **Class** is a blueprint or template that defines what attribute and methods **Objects** can have

- **Charlie and Max** are made from a **Dog** template — **Objects** are made from a **Class** template

- **Class** is a shape with which many individual Objects can be created — in the same way Dog is a shape with which Charlie, Max, etc can be created

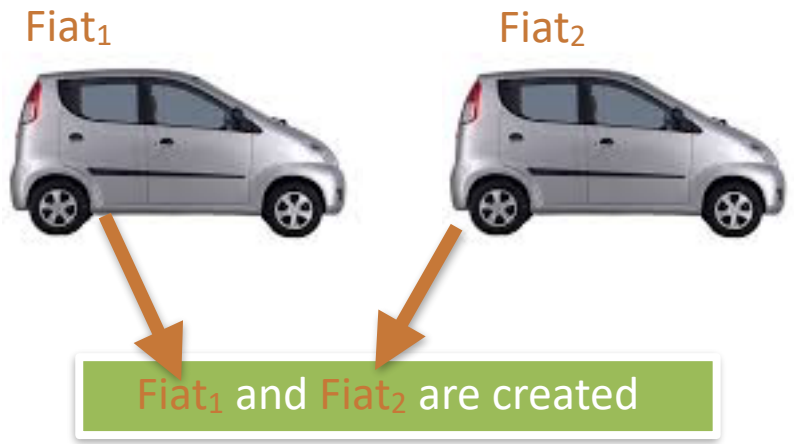
Object Oriented Programming (OOP)



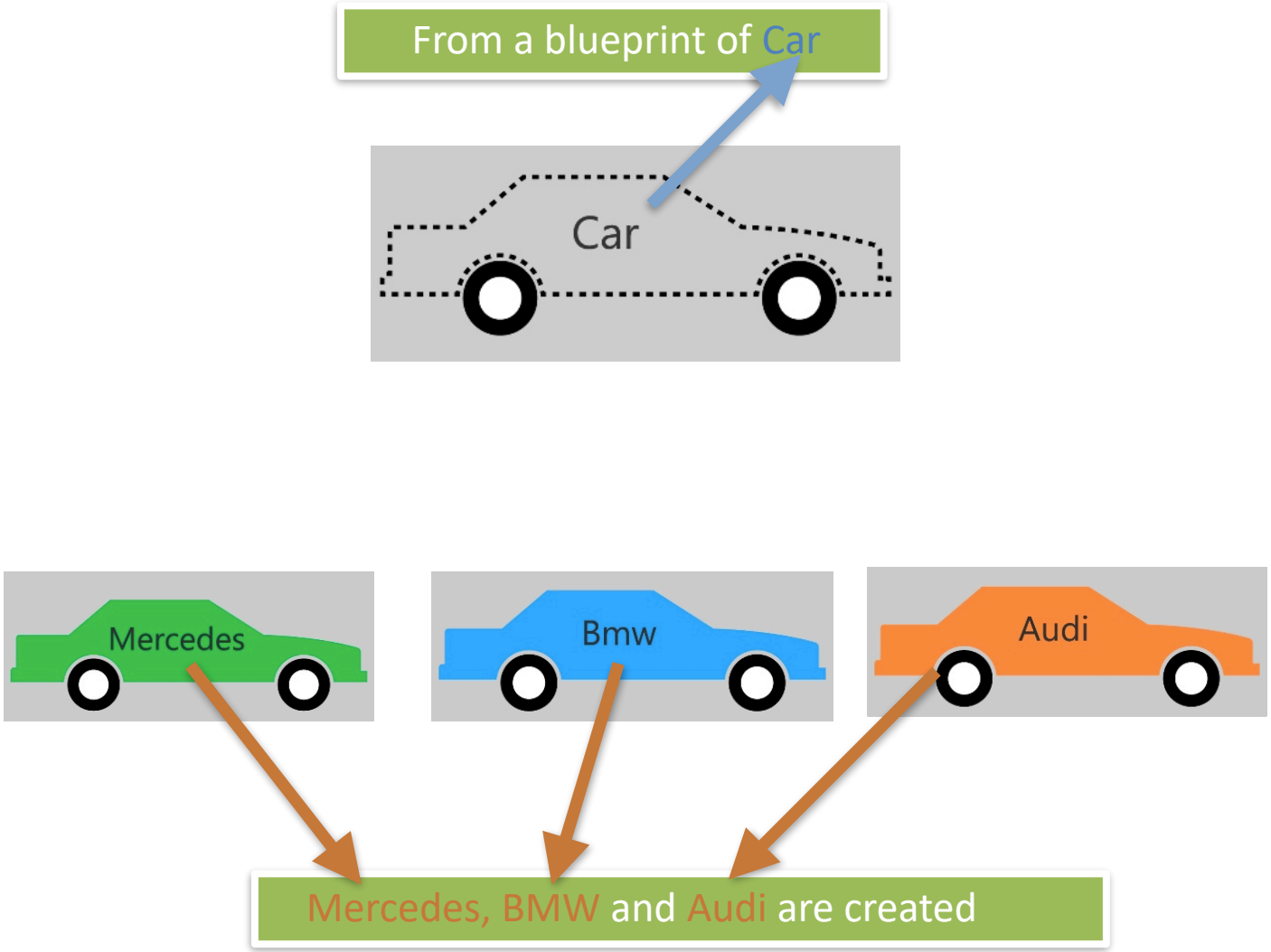
- **Class** is a blueprint or template that defines what attribute and methods **Objects** can have

- **Fiat$_1$** and **Fiat$_2$** made with a **Car** template — **Objects** are made from a **Class** template

- **Class** is a shape with which many individual **Objects** can be created — in the same way Car is a shape with which **Fiat$_1$** and **Fiat$_2$** can be created



Object Oriented Programming (OOP)



Object Oriented Programming (OOP)

From a blueprint of House

House

- **Class** is a blueprint or template that defines what attribute and methods **Objects** can have

- **Sweet Home** and **Corporate Home** are made from a **House** template —
  Objects are made from a **Class** template

Sweet Home

Corporate Home

- **Class** is a shape with which many individual **Objects** can be created — in the same way House is a shape with which **Sweet Home** and **Corporate Home** can be created

Sweet Home and Corporate Home are created