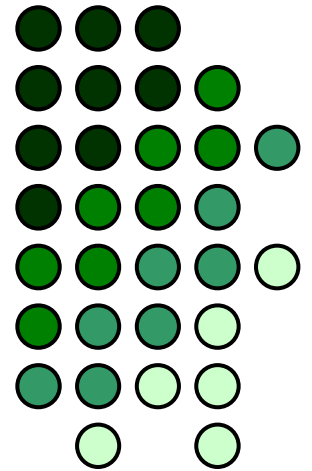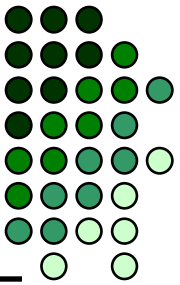# Object Oriented Programming in Java
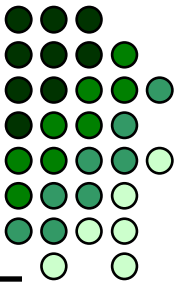
Paul Inventado

De La Salle University

OOP in Java

Classes

- Basic building blocks of an object-oriented language like Java

- Templates that describe the data (or attributes) and behavior (or methods) associated with instances (or objects)

- Referred to as user-defined types or programmer-defined types
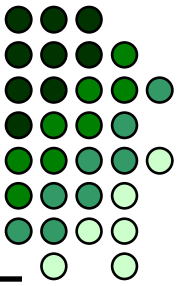
OOP in Java

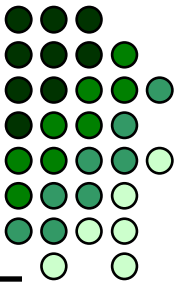| Point |
| --- |
| - x: int |
| - y: int |
| + setPoint(int, int): void |
| + getX(): int |
| + getY(): int |

OOP in Java

[Access Modifier] [Qualifiers] class <Classname>
{
   <class body>
}

- Access Modifiers
  - Specifies the access level of a class
  - Usually public
- Qualifiers
  - abstract or final
- Class name
  - Name of the class

OOP in Java

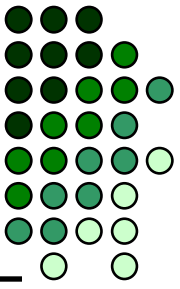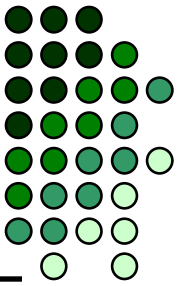| Point |
|---|
| - x: int |
| - y: int |
| + setPoint(int, int): void |
| + getX(): int |
| + getY(): int |

public class Point

{

}

OOP in Java

Attributes

- Used to hold the state of an object of the class
- Referred to as instance variables if non-static
- Referred to as class variables if static
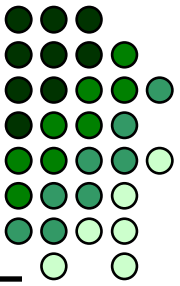
OOP in Java

[Access Modifier][Qualifiers] <data type> <identifier>;

- Access Modifier
  - Specifies the access level of an attribute
  - Usually private
- Qualifiers
  - Abstract, static or final
- Identifier
  - Name of the attribute

OOP in Java
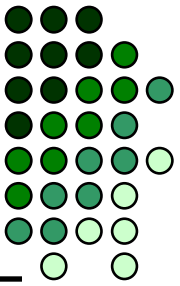
Point
- x: int
- y: int
+ setPoint(int, int): void
+ getX(): int
+ getY(): int

```
public class Point
{
        private int x;
        private int y;

}
```
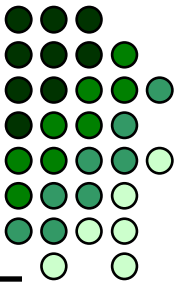
OOP in Java

Methods

- Define the object's behavior
- Public methods are also referred to as public services or public interface of the class
- Public methods are used by clients of the class to manipulate the data stored in objects of the class
- Private methods are known as utility methods or helper methods
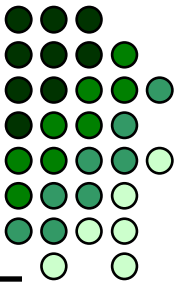
OOP in Java

```
[Access Modifier][Qualifiers] <return type> <identifier>([argument list])
{
    <method body>
}
```
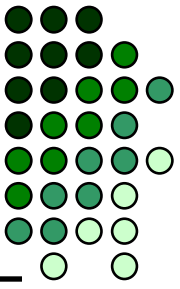
- Access Modifier
  - Specifies the access level of a method
- Qualifiers
  - Abstract, static or final
- Return type
  - Data type to be returned by the method

OOP in Java

- Identifier
  - Name of the method

- Argument List
  - Arguments (or parameters) used by the method

OOP in Java

Point
- x: int
- y: int
+ setPoint(int, int): void
+ getX(): int
+ getY(): int

```java
public class Point
{    private int x;
     private int y;

     public void setPoint(int a, int b)
     {

          x = a;
          y = b;

     }
     public int getX()
     {

          return (x);

     }
     public int getY()
     {

          return (y);

     }
}
```
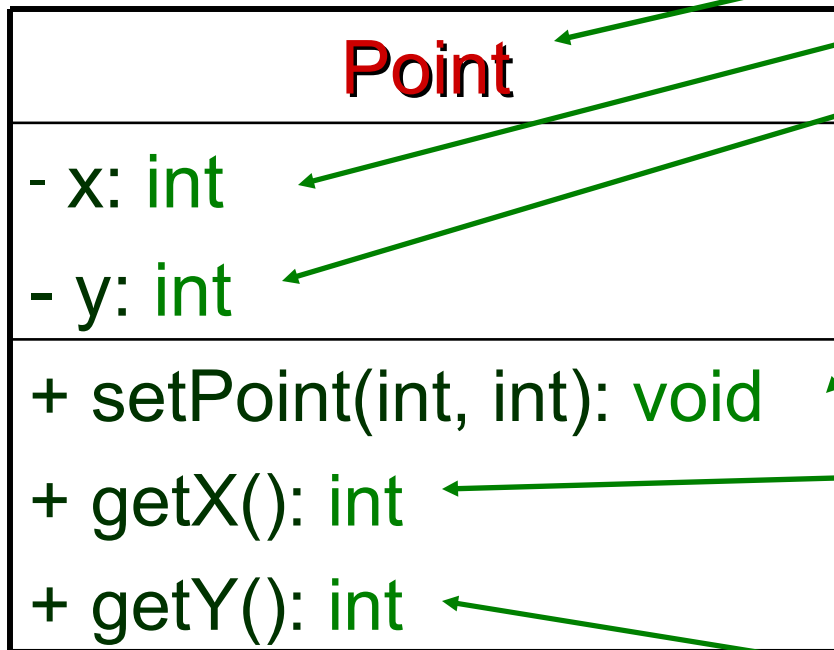
OOP in Java

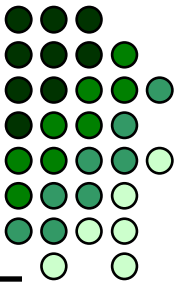| Point |
|---|
| - x: int |
| - y: int |
| + setPoint(int, int): void |
| + getX(): int |
| + getY(): int |

```java
public class Point
{ private int x;
  private int y;

  public void setPoint(int a, int b)
  {
        x = a;
        y = b;
  }
  public int getX()
  {
        return (x);
  }
  public int getY()
  {
        return (y);
  }
}
```

Object Oriented Programming in Java

Paul Inventado

De La Salle University