# Introduction: Polymorphism

- In the previous slide, the two variables are defined to have the same type at compile time: Bank Account
  - However, the types of objects they are referring to at runtime are different
- What happens when the withdraw method is invoked on each object?
  - anAccount refers to an instance of BankAccount. Therefore, the withdraw method defined in BankAccount is invoked.
  - account1 refers to an instance of OverdraftAccount. Therefore, the withdraw method defined in OverdraftAccount is invoked.
- Polymorphism is: The method being invoked on an object is determined AT RUNTIME and is based on the type of the object receiving the message.

# Final Methods and Final Classes

- Methods can be qualified with the final modifier
  - Final methods cannot be overridden.
  - This can be useful for security purposes.

```
public final boolean validatePassword(String username, String Password)
{
    [...]
```

- Classes can be qualified with the final modifier
  - The class cannot be extended
  - This can be used to improve performance. Because there an be no subclasses, there will be no polymorphic overhead at runtime.

```
public final class Color
{
    [...]
```

Scope: Program for Implementing Polymorphism

To retrieve data with the help of object.

```
Interface A
{
}
Class B implements A
{
Public static void main ()
{
A b = new B();
}
}
```