

Java Programming 2-3: Java Class Design –

Abstract Classes Practice Activities

JavaBank.java Updates:

- a. AbstractBankAccount
- b. displayAccountDetails(AbstractBankAccount account)
- c. account.toString()
- d. Ensure method calls pass AbstractBankAccount objects.

Reason for Abstract Class: Abstract classes can provide common behavior (methods) with some implementation, while interfaces cannot.

Update Bike Class: Make Bike abstract: public abstract class Bike.

Remove bike4 Code: Delete or comment out bike4 instantiation from the driver class.

Convert printDescription to toString: Change method name to toString() in all classes, then update console display code

```
public class Animal {  
    public void makeNoise() {  
        System.out.println("talk");  
    } //end method makeNoise  
} //end class Animal
```

```
public class Dog extends Animal {  
    public void makeNoise() {  
        System.out.println("Bark");  
    } //end method makeNoise  
} //end class Dog
```

```

Animal animal = new Animal();
animal.makeNoise();
Dog dog = new Dog();
dog.makeNoise();
Animal animaldog = new Dog();
animaldog.makeNoise();

```

Output:

talk
 Bark
 Bark
 The makeNoise method in Dog overrides the one in Animal, so Dog instances use Bark.

```

if (animal instanceof Animal)
    System.out.println("animal is Animal");
if (dog instanceof Animal)
    System.out.println("dog is Animal");
if (animaldog instanceof Animal)
    System.out.println("animaldog is Animal");
if (animal instanceof Dog)
    System.out.println("animal is Dog");

```

B.

Output:

animal is Animal
 dog is Animal
 animaldog is Animal

animal and dog are instances of Animal and dog respectively, while animaldog is an instance of Dog, which is a subclass of Animal. animal is not a Dog.

7.Casting Description:

Primitives: Convert between data types, e.g., int to double (automatic) or double to int (explicit).

Objects: Change object type reference, e.g., upcasting to a superclass or downcasting to a subclass.

8.Casting Examples:

Upcast: `Animal animal = new Dog();`

Downcast: `Dog dog = (Dog) animaldog;` (after checking instanceof)