Assigned in: Lesson 3

Due before: Lesson 4

For this lab we will create a hierarchy of several classes: IDevice (the parent), IPod, IPad, and IPhone.

The parent class – IDevice – has a concrete method called **getPurpose()** which returns its purpose (a String). The "purpose" instance variable is set in the constructor. The IDevice class also has an abstract method called **printDetails()** which prints all of the child class's instance variables.

Child classes of IDevice:

- IPod:	the purpose of this iDevice is "music"
- IPad:	the purpose of this iDevice is "learning"
- IPhone:	the purpose of this iDevice is "talking"

Note: the child classes also contain instance variables, constructor parameters, accessor methods, and mutator methods for several other data members:

- IPod:	(int) number of songs stored, (double) maximum volume in decibels
- IPad:	(boolean) has a case, (String) operating system version
- IPhone:	(double) number of minutes remaining on phone plan, (String) carrier

Also, each of these four classes also overrides the **toString()** method to return all of the instance variables in a single String. Use the @Override annotation. Child classes' **toString()** methods must also call their parent's toString() method.

Furthermore, each of these four classes also overrides the **equals()** method. IPods with the same number of songs stored are considered equal; IPads with the same operating system version are considered equal; IPhones which have the same amount of minutes remaining as each other are considered equal. Use the @Override annotation when overriding **equals()**.

Remember to override the **hashCode()** method too for each class, whenever you override **equals()**.

Continuing from above, add more data and methods as described below.

Extend the IPhone class; it has a child called IPhone16.

The IPhone16 class also contains instance variables, constructor parameters, accessor methods, and mutator methods for several other data members:

- (boolean) high-resolution camera
- (int) gigabytes of memory

Also, the IPhone16 class overrides the **toString()** method to return all of the object data in a String. Use the @Override annotation. This **toString()** method must also call its parent's **toString()** method.

Furthermore, this class also overrides **equals()** (and therefore also **hashCode()**). IPhone16 objects that have the same amount of minutes remaining on their phone plan are considered equal, but only if they also have the same value for "high-resolution camera". Use the @Override annotation.

Remember to override **hashCode()** properly too for this class.

Create a Main class with the main() method. This method creates three objects of each of your four classes, and tells if your code looks correct or not (copy/paste this code BUT ALSO FIX ALL THE STYLE VIOLATIONS):

```
public class Main {
  public static void main(final String[] args) {
    // Create IPod objects
    final IPod ipod1;
    final IPod ipod2;
    final IPod ipod3;
    ipod1 = new IPod(300, 80.0);   // 300 songs, max volume 80.0 dB
    ipod2 = new IPod(400, 85.0);   // 400 songs, max volume 85.0 dB
    ipod3 = new IPod(300, 70.0);   // 300 songs, max volume 70.0 dB

    // Test equality and inequality for IPod
    System.out.println("IPod Equality Test:");
    if (!ipod1.equals(ipod2)) {
      System.out.println("CORRECT: ipod1 is not equal to ipod2");
    } else {
      System.out.println("INCORRECT: ipod1 should not be equal to ipod2");
    }

    if (ipod1.equals(ipod3)) {
      System.out.println("CORRECT: ipod1 is equal to ipod3");
    } else {
      System.out.println("INCORRECT: ipod1 should be equal to ipod3");
    }
    System.out.println();

    // Create IPad objects
    final IPad ipad1;
    final IPad ipad2;
    final IPad ipad3;
    ipad1 = new IPad(true, "iPadOS 15");   // Has case, OS version iPadOS 15
    ipad2 = new IPad(false, "iPadOS 14");  // No case, OS version iPadOS 14
    ipad3 = new IPad(true, "iPadOS 15");   // Has case, OS version iPadOS 15

    // Test equality and inequality for IPad
    System.out.println("IPad Equality Test:");
    if (!ipad1.equals(ipad2)) {
      System.out.println("CORRECT: ipad1 is not equal to ipad2");
    } else {
      System.out.println("INCORRECT: ipad1 should not be equal to ipad2");
    }
```

```
if (ipad1.equals(ipad3)) {
  System.out.println("CORRECT: ipad1 is equal to ipad3");
} else {
  System.out.println("INCORRECT: ipad1 should be equal to ipad3");
}
System.out.println();

// Create IPhone objects
final IPhone iphone1;
final IPhone iphone2;
final IPhone iphone3;
iphone1 = new IPhone(120.0, "Verizon");   // 120 minutes, carrier Verizon
iphone2 = new IPhone(180.0, "T-Mobile");  // 180 minutes, carrier T-Mobile
iphone3 = new IPhone(120.0, "AT&T");      // 120 minutes, carrier AT&T

// Test equality and inequality for IPhone
System.out.println("IPhone Equality Test:");
if (!iphone1.equals(iphone2)) {
  System.out.println("CORRECT: iphone1 is not equal to iphone2");
} else {
  System.out.println("INCORRECT: iphone1 should not be equal to iphone2");
}

if (iphone1.equals(iphone3)) {
  System.out.println("CORRECT: iphone1 is equal to iphone3");
} else {
  System.out.println("INCORRECT: iphone1 should be equal to iphone3");
}
System.out.println();

// Create IPhone16 objects
final IPhone16 iphone16_1;
final IPhone16 iphone16_2;
final IPhone16 iphone16_3;
iphone16_1 = new IPhone16(100.0, "Verizon", true, 512);   // 100 minutes, high-res camera, 512 GB
iphone16_2 = new IPhone16(100.0, "Verizon", true, 256);   // 100 minutes, high-res camera, 256 GB
iphone16_3 = new IPhone16(100.0, "Verizon", false, 512);  // 100 minutes, no high-res camera, 512 GB

// Test equality and inequality for IPhone16
System.out.println("IPhone16 Equality Test:");
if (iphone16_1.equals(iphone16_2)) {
  System.out.println("CORRECT: iphone16_1 is equal to iphone16_2");
} else {
  System.out.println("INCORRECT: iphone16_1 should be equal to iphone16_2");
}

if (!iphone16_1.equals(iphone16_3)) {
```

```
            System.out.println("CORRECT: iphone16_1 is not equal to iphone16_3");
        } else {
            System.out.println("INCORRECT: iphone16_1 should not be equal to iphone16_3");
        }
        System.out.println();
    }
}
```