Below are some review notes based on some FAQs.

1. **Difference between class and object:**

   An object is simply a variable that is not a primitive data type (int, boolean, char, double). For example, a String variable, a BankAccount variable, a Hangman game, and any array are all examples of objects. A class defines the properties (data variables and methods of an object). The String class and array class are built into the Java language, whereas we wrote the BankAccount and Hangman classes. The class is the code here (for Bank Account) and here (for Hangman):

   In the following statement:

   ```java
   BankAccount acct = new BankAccount("12345");
   ```

   `acct` is the object and `BankAccount` is the class.

2. **Global variables:**

   A global variable is one that can be accessed by the entire class that is declared in. To make a variable global, declare it somewhere outside of a method. If it is in the Driver class, the variable must be declared **static**.

   For example, for a Driver program, we would declare a global variable as follows:

   ```java
   import java.util.Scanner;
   public class Driver{
       static Scanner scan = new Scanner(System.in); //global
       public static void main(...)
       {
           int x = scan.nextInt(); //accessing a
           //global variable
       }

   ...
   ```

   For a class program, we would declare a global variable as follows:

   ```java
   public class Hangman{
       String[] words; //global variable

       public String[] getWords)
       {
           return this.words; //accessing a global variable
   ```
Notice in each case, the global variable is not declared in any method.

3. **Method calls:**

To call a method that is within the same class, we don’t need to use dot notation.

- From within BankAccount class:
  
  ```java
  deposit(100);
  ```

- From within a Driver class:
  
  ```java
  getAverage();
  ```

To call a method from the Driver class to an object class:

```java
Hangman game = new Hangman();
game.getWords();
```

Method calls are not made from an object class to a Driver class.

4. **private vs. public** (applies to both variables and methods):

`private` and `public` are visibility modifiers that enforce the accessibility of a variable or method. We use these modifiers for object (not Driver) classes.

- `private` – accessible from only within the same class
- `public` – accessible from any class in the same directory

For example:

```java
public class BankAccount{

    private String acctNum; //directly accessible
    //from only this class

    ...

    For an outside (e.g. Driver) class to access acctNum, we need getter/setter method in BankAccount (see #5).

5. **Setters and Getters:**

```java
public class BankAccount{
```
private String acctNum; //directly accessible
       //from only this class

...

//Setter for acctNum
public void setAcctNum(String num)
{
    this.acctNum = num;
}

//Getter for acctNum
public String getAcctNum()
{
    return this.acctNum;
}

...

Then in the Driver, we would access acctNum as follows:

public class Driver{

    public static void main(String[] args){
    {
        BankAccount acct = new BankAccount("12345");
        acct.setAcctNum("12345");

        ... System.out.println(acct.getAcctNum());
    }

    ...
}