1. Extend the Bicycle class by adding the second data member tagNo of type String. Declare this data member as private.

```java
class Bicycle {

   // Data Members
   private String ownerName;
   private String tagNo;

   //Constructor: Initializes the data member
   public void Bicycle() {
      ownerName = "Unknown";
      tagNo = "Unassigned";
   }

   ...
}
```

2. Add a new method to the Bicycle class that assigns a tag number. This method will be called as

```java
Bicycle bike;
bike = new Bicycle();
```
Solutions to Chapter 4 Quick Check Questions

... bike.setTagNo("2004-134R");

Answer

class Bicycle {
    ...
    public void setTabNo(String number) {
        tagNo = number;
    }
    ...
}

3. Add a another method to the Bicycle class that returns the bicycle’s tag number. This method will be called a

Bicycle bike;
bike = new Bicycle();
...
String tag = bike.getTagNo();

Answer

class Bicycle {
    ...
    public String getTagNo() {
        return tagNo;
    }
    ...
}

4.2 Second Example: Defining and Using Multiple Classes

1. What is the output from the following code fragment?

Account acct;
acct = new Account();
acct.setInitialBalance(250);
acct.add(20);
System.out.println("Balance: "+ acct.getCurrentBalance());

Answer:

Balance: 270

2. Write a code fragment to declare and create two Account objects named acc1 andacct2. Initialize the balance to $300 and $500, respectively. Set the name of owner for both accounts to John Doe.

Answer:

Account acct1, acct2;
acct1 = new Account();
acct2 = new Account();
acct1.setInitialBalance(300);
acct2.setInitialBalance(300);

4.3 Matching Arguments and Parameters

No Quick Check Questions.

4.4 Passing Objects to a Method

No Quick Check Questions.

4.5 Constructors

1. Which of the following constructors are invalid?

   public int ClassA(int one) { <--- Invalid
   ...
   }

   ...
Solutions to Chapter 4 Quick Check Questions

```java
public class B(int one, int two) {
    ...
}

void classC() {
    ...
}

1. Invalid. The constructor does not have a return type.
2. Valid.
3. Invalid. The void modifier is not valid with the constructor.

2. What is the main purpose of a constructor?

The main purpose of a constructor is to initialize the data members of a class fully so an instance will be created in a valid state.

3. Complete the following constructor:

```java
class Test {
    private double score;

    public Test(double val) {
        //assign the value of parameter to //the data member
        score = val;
    }
}
```

Answer

```java
class Test {
    private double score;

    public Test(double val) {
        score = val;
    }
}
```
4.6 Information Hiding and Visibility

1. If the data member speed is private, is the following statement valid in a client program?

   ```java
   Robot aibo;
   aibo = new Robot();
   double currentSpeed = aibo.speed;  // Invalid
   ```

   *No, the statement is invalid. You do not have a direct access to the private data members of a class from a client code.*

2. Suppose you wrote down important information such as your bank account number, student registration ID, and so forth, on a single sheet of paper. Will this sheet be declared private, and kept in your desk drawer, or public, and placed next to the dorm’s public telephone?

   *Confidential information should be hidden from the public view or access, so the sheet should be declared private and hidden from the public.*

3. Identify the private methods from the following diagram:

   ```
   MyClass
   - mydata : double
   + MyClass( )
     + methodOne(double) : void
     - methodTwo(double) : double
     - methodThree(double) : double
   ```

   *Private methods are methodTwo and methodThree.*
4.7 Class Constants

1. Declare two class constants named MIN_BALANCE and MAX_BALANCE whose data types are double.

   public static final double MIN_BALANCE = 100.0;
   public static final double MAX_BALANCE = 500.0;

2. Is there any problem with the following declarations?

   class Question {
       private final int MAX = 20;
       ...
   }

   *The static modifier is missing in the declaration.*

3. Modify the Dice class so its instances will generate a number between 5 and 15, inclusively.

   class Dice {
       //Data Members
       private static final int MAX_NUMBER = 15;
       private static final int MIN_NUMBER = 5;

       //the rest is the same
       ...
   }

4.8 Local Variables

1. How is a local variable different from an instance variable?

   *Local variables are temporary variables used only by the method in which they are declared. Memory space for local variables exist only during the execution of the method.*

2. Rewrite the following method using local variables:
public int totalCharge(int amt) {
    return (balance - (int) Math.round(amt * 1.5));
}

Answer
public int totalCharge(int amt) {
    int result = balance - (int) Math.round(amt * 1.5);
    return result;
}

4.9 Calling Methods of the Same Class

1. Suppose a class Alpha includes a method called compute that accepts no arguments. Define another method of Alpha named myMethod that calls the compute method.

    class Alpha {
        ...
        public void myMethod() {
            compute();
        }
        ...
    }

2. Why should duplication of code be avoided?

    Duplication of code should be avoided as much as possible to streamline the modification process and make the process less error-prone.

4.10 Changing Any Class to a Main Class

    No Quick Check Questions
Solutions to Chapter 4 Quick Check Questions

4.11 Sample Development: Loan Calculator

No Quick Check Questions