

1. This question involves dog walkers who are paid through a dog-walking company to take dogs for one-hour walks. A dog-walking company has a varying number of dogs that need to be walked during each hour of the day. A dog-walking company is represented by the following `DogWalkCompany` class.

```
public class DogWalkCompany
{
    /**
     * Returns the number of dogs, always greater than 0, that are available
     * for a walk during the time specified by hour
     * Precondition: 0 <= hour <= 23
     */
    public int numAvailableDogs(int hour)
    { /* implementation not shown */ }

    /**
     * Decreases, by numberDogsWalked, the number of dogs available for a walk
     * during the time specified by hour
     * Preconditions: 0 <= hour <= 23
     *                 numberDogsWalked > 0
     */
    public void updateDogs(int hour, int numberDogsWalked)
    { /* implementation not shown */ }

    /* There may be instance variables, constructors,
       and methods that are not shown. */
}
```

A dog walker is associated with a dog-walking company and is represented by the `DogWalker` class. You will write two methods of the `DogWalker` class.

```
public class DogWalker
{
    /** The maximum number of dogs this walker can walk simultaneously
        per hour */
    private int maxDogs;

    /** The dog-walking company this dog walker is associated with */
    private DogWalkCompany company;

    /**
     * Assigns max to maxDogs and comp to company
     * Precondition: max > 0
     */
    public DogWalker(int max, DogWalkCompany comp)
    { /* implementation not shown */ }

    /**
     * Takes at least one dog for a walk during the time specified by
     * hour, as described in part (a)
     * Preconditions: 0 <= hour <= 23
     *                 maxDogs > 0
     */
    public int walkDogs(int hour)
    { /* to be implemented in part (a) */ }
```

```
/**
 * Performs an entire dog-walking shift and returns the amount
 * earned, in dollars, as described in part (b)
 * Preconditions: 0 <= startHour <= endHour <= 23
 *                maxDogs > 0
 */
public int dogWalkShift(int startHour, int endHour)
{ /* to be implemented in part (b) */ }

/* There may be instance variables, constructors,
   and methods that are not shown. */
}
```

- A. Write the `walkDogs` method, which updates and returns the number of dogs this dog walker walks during the time specified by `hour`. Values of `hour` range from 0 to 23, inclusive.

A helper method, `numAvailableDogs`, has been provided in the `DogWalkCompany` class. The method returns the number of dogs available to be taken for a walk at a given hour. The dog walker will always walk as many dogs as the dog-walking company has available to walk, as long as the available number of dogs is not greater than the maximum number of dogs that the dog walker can handle, represented by the value of `maxDogs`.

Another helper method, `updateDogs`, has also been provided in the `DogWalkCompany` class. So that multiple dog walkers do not sign up to walk the same dogs, the `walkDogs` method should use `updateDogs` to update the dog-walking company with the number of dogs this dog walker will walk during the given hour. The parameters of the `updateDogs` method indicate how many dogs this dog walker will walk during the time specified by `hour`.

For example, if the dog-walking company has 10 dogs that need to be walked at the given hour but the dog walker's maximum is 4, the `updateDogs` method should be used to indicate that this dog walker will walk 4 of the 10 dogs during the given hour. As another example, if the dog-walking company has 3 dogs that need to be walked at the given hour and the dog walker's maximum is 4, the `updateDogs` method should be used to indicate that this dog walker will walk all 3 of the available dogs during the given hour.

The `walkDogs` method should return the number of dogs to be walked by this dog walker during the time specified by `hour`.

Complete method `walkDogs`. You must use `numAvailableDogs` and `updateDogs` appropriately to receive full credit.

```
/**
 * Takes at least one dog for a walk during the time specified by
 * hour, as described in part (a)
 * Preconditions: 0 <= hour <= 23
 *                maxDogs > 0
 */
public int walkDogs(int hour)
```

- B. Write the `dogWalkShift` method, which performs a dog-walking shift of the hours in the range `startHour` to `endHour`, inclusive, and returns the total amount earned. For example, a dog-walking shift from 14 to 16 is composed of three one-hour dog walks starting at hours 14, 15, and 16.

For each hour, the base pay is \$5 per dog walked plus a bonus of \$3 if at least one of the following is true.

- `maxDogs` dogs are walked
- the walk occurs between the peak hours of 9 and 17, inclusive

The following table shows an example of the calculated amount earned by walking dogs from hour 7 through hour 10, inclusive.

Hour	Maximum Number of Dogs	Number of Dogs Walked	Amount Earned, in Dollars
7	3	3	$3 \times 5 + 3 = 18$
8	3	2	$2 \times 5 = 10$
9	3	2	$2 \times 5 + 3 = 13$
10	3	3	$3 \times 5 + 3 = 18$
Total			59

Complete method `dogWalkShift`. Assume that `walkDogs` works as specified, regardless of what you wrote in part (a). You must use `walkDogs` appropriately to earn full credit.

```
/**
 * Performs an entire dog-walking shift and returns the amount
 * earned, in dollars, as described in part (b)
 * Preconditions: 0 <= startHour <= endHour <= 23
 *                maxDogs > 0
 */
public int dogWalkShift(int startHour, int endHour)
```