

Tracing Objects

Object Memory Diagram

Coding Your Own Object



```
public class CrewMate {  
    private String colour;  
    private String hat;  
    private int tasksDone;  
  
    public CrewMate() {  
        colour="red";  
        hat="no";  
        tasksDone=0;  
    }  
    public CrewMate(String c, String h) {  
        colour=c;  
        hat=h;  
        tasksDone=0;  
    }  
    public int getTasksDone() {  
        return tasksDone;  
    }  
    public void completeTasks() {  
        tasksDone++;  
    }  
    public void setTasksDone(int n) {  
        tasksDone=n;  
    }  
    public String toString() {  
        return colour+": "+hat+" hat, " +  
            tasksDone + " tasks";  
    }  
}
```

Instance Variables.
(The data)

Constructors,
Mutators,
Accessors,
Facilitators.
(The methods)

Classify each method.



```
public class CrewMate {  
    private String colour;  
    private String hat;  
    private int tasksDone;
```

```
    public CrewMate() {  
        colour="red";  
        hat="no";  
        tasksDone=0;  
    }  
    public CrewMate(String c, String h) {  
        colour=c;  
        hat=h;  
        tasksDone=0;  
    }  
    public int getTasksDone() {  
        return tasksDone;  
    }  
    public void completeTasks() {  
        tasksDone++;  
    }  
    public void setTasksDone(int n) {  
        tasksDone=n;  
    }  
    public String toString() {  
        return colour+": "+hat+" hat, " +  
            tasksDone + " tasks";  
    }  
}
```

1

2

3

4

5

6

Classify each method.



```
public class CrewMate {  
    private String colour;  
    private String hat;  
    private int tasksDone;
```

```
    public CrewMate() {  
        colour="red";  
        hat="no";  
        tasksDone=0;  
    }  
    public CrewMate(String c, String h) {  
        colour=c;  
        hat=h;  
        tasksDone=0;  
    }  
    public int getTasksDone() {  
        return tasksDone;  
    }  
    public void completeTasks() {  
        tasksDone++;  
    }  
    public void setTasksDone(int n) {  
        tasksDone=n;  
    }  
    public String toString() {  
        return colour+": "+hat+" hat, " +  
            tasksDone + " tasks";  
    }  
}
```

1 Constructor

2 Constructor

3 Accessor

4 Facilitator

5 Mutator

6 Accessor

Objects need to be declared in another class.

When you were using TextViews and Buttons, this is the class that you wrote.

```
public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
        CrewMate nothingSpecial = new CrewMate ();
        CrewMate egghead = new CrewMate ("yellow", "egg");
        System.out.println(nothingSpecial);
        System.out.println(egghead);
        egghead.completeTasks ();
        nothingSpecial.setTasksDone (5);
        nothingSpecial.completeTasks ();
        System.out.println(egghead.getTasksDone ());
        System.out.println(nothingSpecial.getTasksDone ());
        egghead.completeTasks ();
        System.out.println(nothingSpecial);
        System.out.println(egghead);
    }
}
```

What are the names of the two variables?

What lines are they declared on?

```
public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

```

What mutator
method is
called?

What line is it
called on?

```
public class AmongUs {  
    public static void main(String args[]) {  
        new AmongUs ();  
    }  
    public AmongUs () {  
1   CrewMate nothingSpecial = new CrewMate ();  
2   CrewMate egghead = new CrewMate ("yellow", "egg");  
3   System.out.println(nothingSpecial);  
4   System.out.println(egghead);  
5   egghead.completeTasks ();  
6   nothingSpecial.setTasksDone (5);  
7   nothingSpecial.completeTasks ();  
8   System.out.println(egghead.getTasksDone ());  
9   System.out.println(nothingSpecial.getTasksDone ());  
10  egghead.completeTasks ();  
11  System.out.println(nothingSpecial);  
12  System.out.println(egghead);  
    }  
}
```

toString is called four times in this class.

What lines is it called on?

```
public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

```


The accessor
getTasksDone
is called
twice.

What lines is
it called on?

```
public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

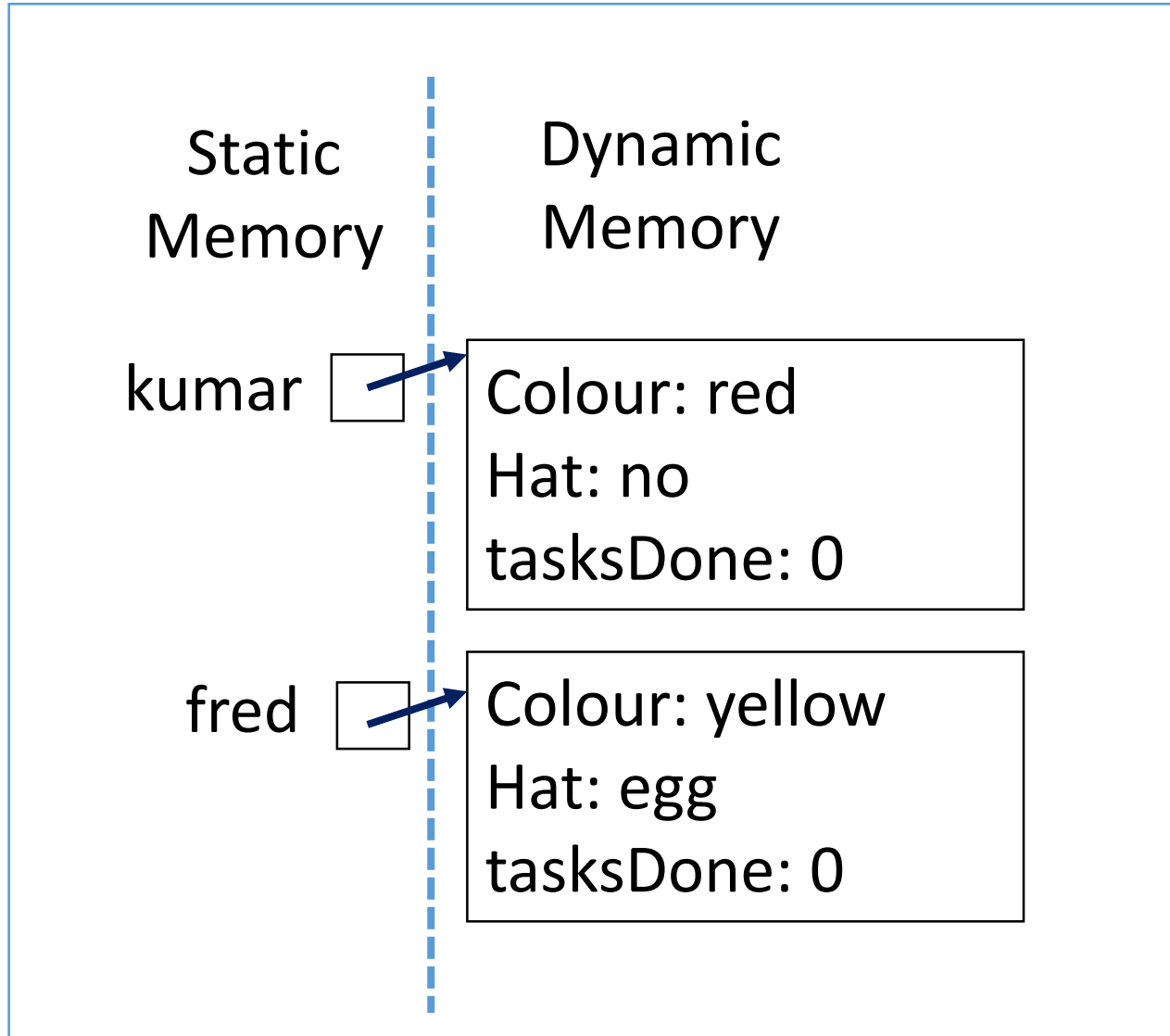
```

Which lines
would print
something on
the screen?

```
public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

```

An Object Memory Diagram



Static Memory

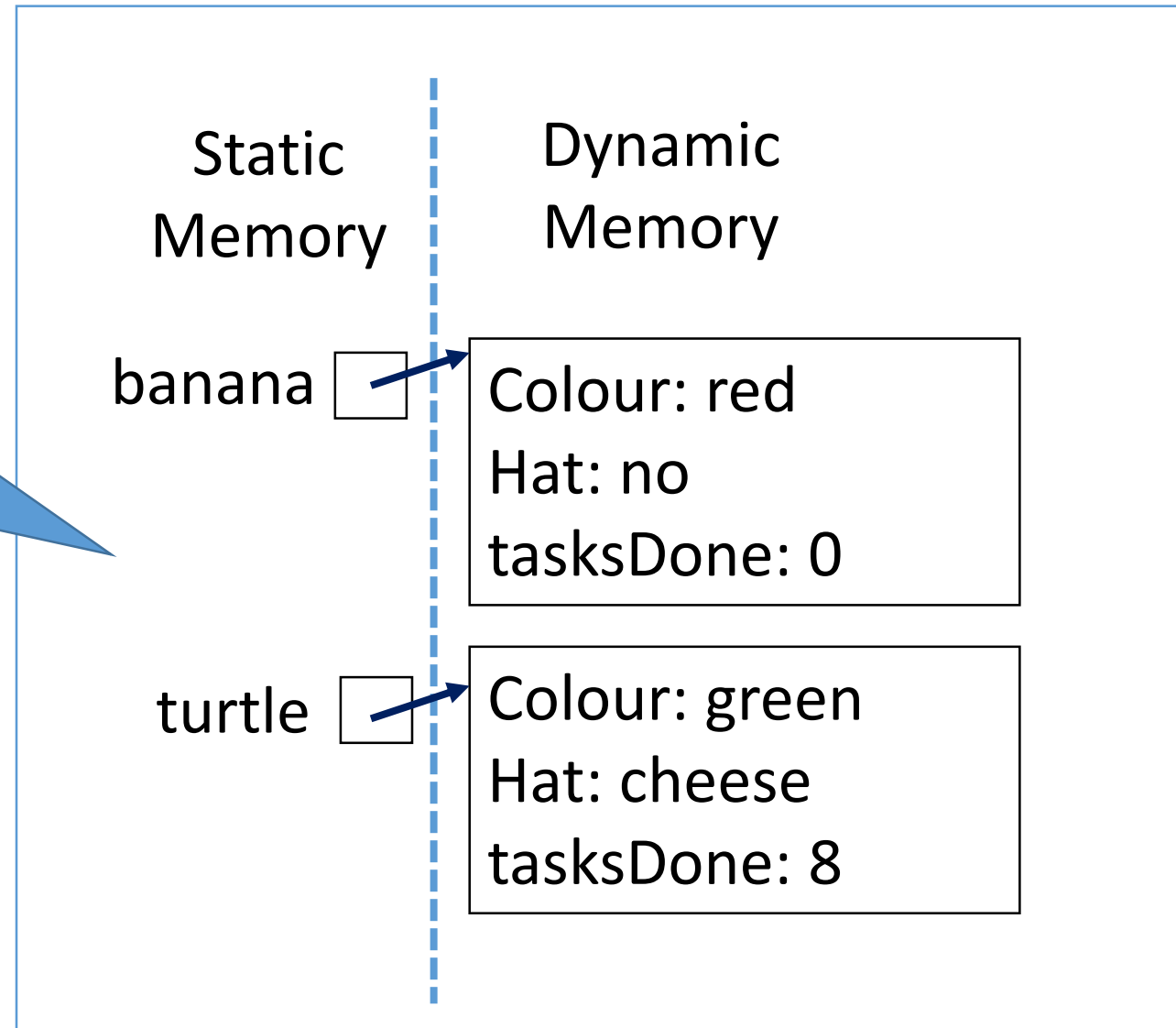
- Allocated by the number of variables you declare.
- Primitive types go here.
- The object names go here.

Dynamic Memory

- Set up and allocated with the keyword 'new'.
- The object's data go here.

An Object Memory Diagram

Two variables (objects).
What are the variable
names?



```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println (nothingSpecial);
4   System.out.println (egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println (egghead.getTasksDone ());
9   System.out.println (nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println (nothingSpecial);
12  System.out.println (egghead);
    }
}

```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Static

Dynamic



Colour:
Hat:
tasksDone:



Colour:
Hat:
tasksDone:

Console


```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1 CrewMate nothingSpecial = new CrewMate ();
2 CrewMate egghead = new CrewMate ("yellow", "egg");
3 System.out.println(nothingSpecial);
4 System.out.println(egghead);
5 egghead.completeTasks ();
6 nothingSpecial.setTasksDone (5);
7 nothingSpecial.completeTasks ();
8 System.out.println(egghead.getTasksDone ());
9 System.out.println(nothingSpecial.getTasksDone ());
10 egghead.completeTasks ();
11 System.out.println(nothingSpecial);
12 System.out.println(egghead);
    }
}

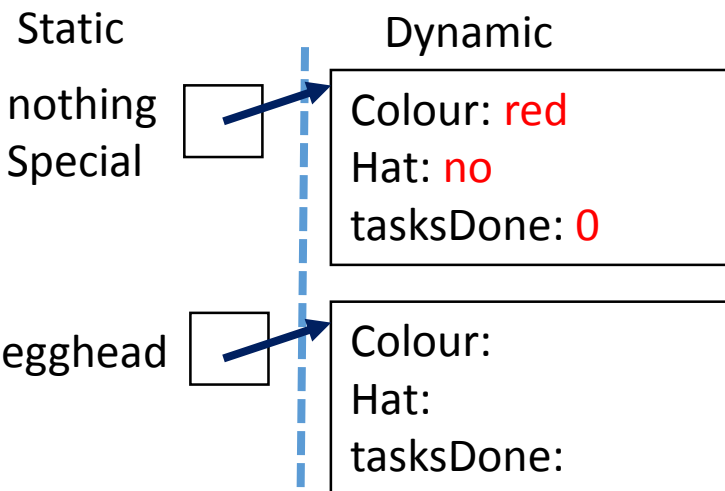
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Console

```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1 CrewMate nothingSpecial = new CrewMate ();
2 CrewMate egghead = new CrewMate ("yellow", "egg");
3 System.out.println(nothingSpecial);
4 System.out.println(egghead);
5 egghead.completeTasks ();
6 nothingSpecial.setTasksDone (5);
7 nothingSpecial.completeTasks ();
8 System.out.println(egghead.getTasksDone ());
9 System.out.println(nothingSpecial.getTasksDone ());
10 egghead.completeTasks ();
11 System.out.println(nothingSpecial);
12 System.out.println(egghead);
    }}

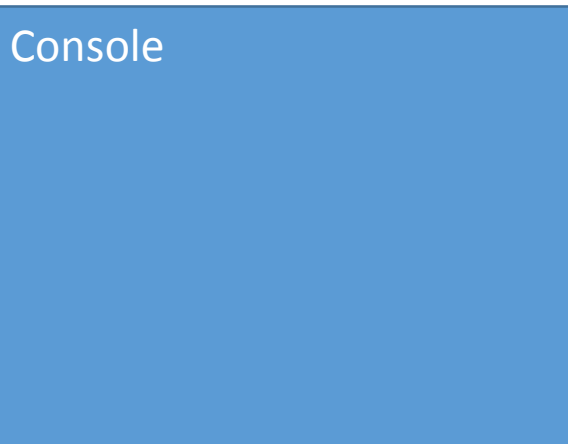
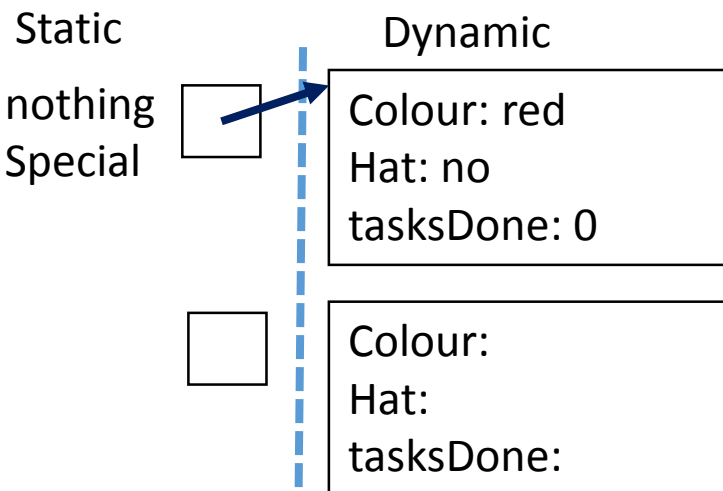
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```




```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1 CrewMate nothingSpecial = new CrewMate ();
2 CrewMate egghead = new CrewMate ("yellow", "egg");
3 System.out.println(nothingSpecial);
4 System.out.println(egghead);
5 egghead.completeTasks ();
6 nothingSpecial.setTasksDone (5);
7 nothingSpecial.completeTasks ();
8 System.out.println(egghead.getTasksDone ());
9 System.out.println(nothingSpecial.getTasksDone ());
10 egghead.completeTasks ();
11 System.out.println(nothingSpecial);
12 System.out.println(egghead);
    }}

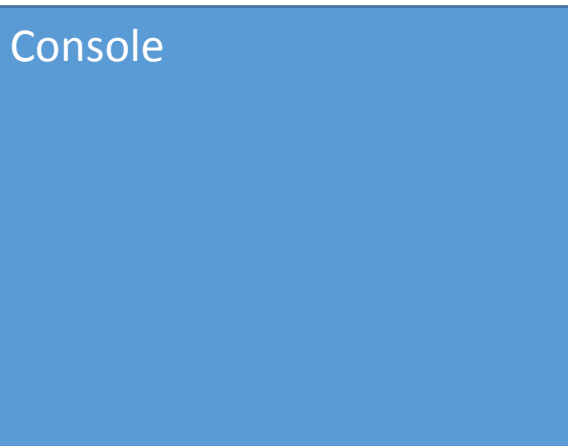
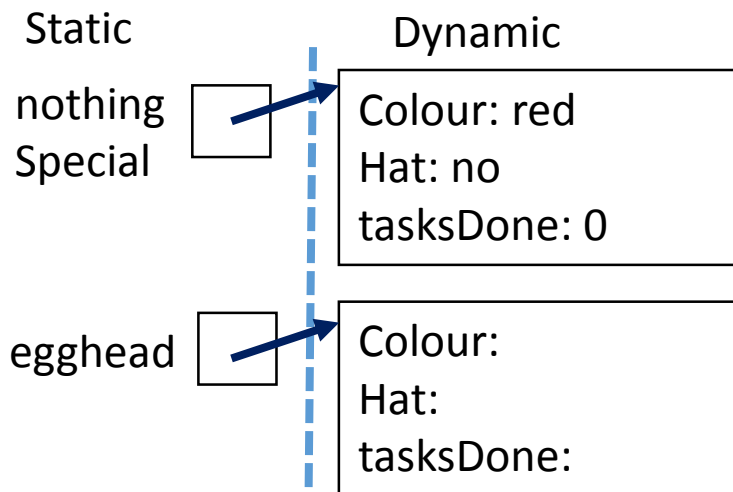
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1 CrewMate nothingSpecial = new CrewMate ();
2 CrewMate egghead = new CrewMate ("yellow", "egg");
3 System.out.println(nothingSpecial);
4 System.out.println(egghead);
5 egghead.completeTasks ();
6 nothingSpecial.setTasksDone (5);
7 nothingSpecial.completeTasks ();
8 System.out.println(egghead.getTasksDone ());
9 System.out.println(nothingSpecial.getTasksDone ());
10 egghead.completeTasks ();
11 System.out.println(nothingSpecial);
12 System.out.println(egghead);
    }}

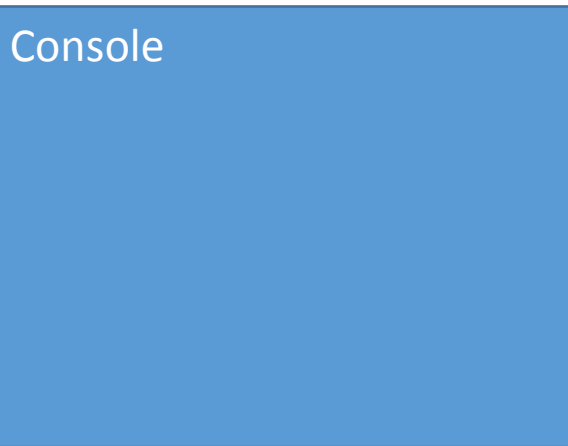
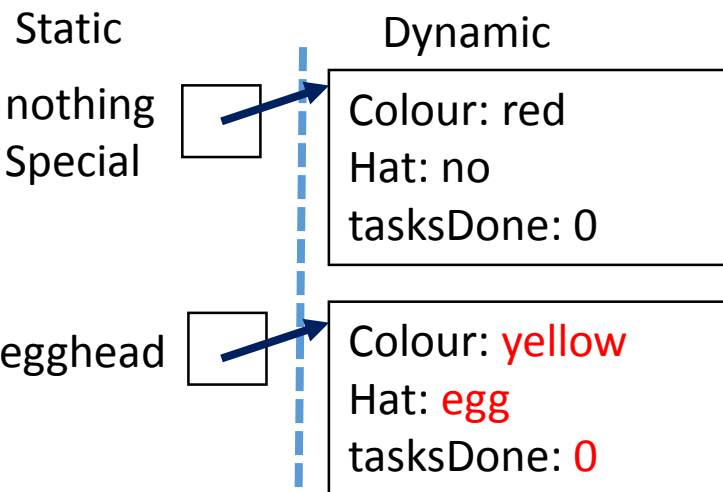
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

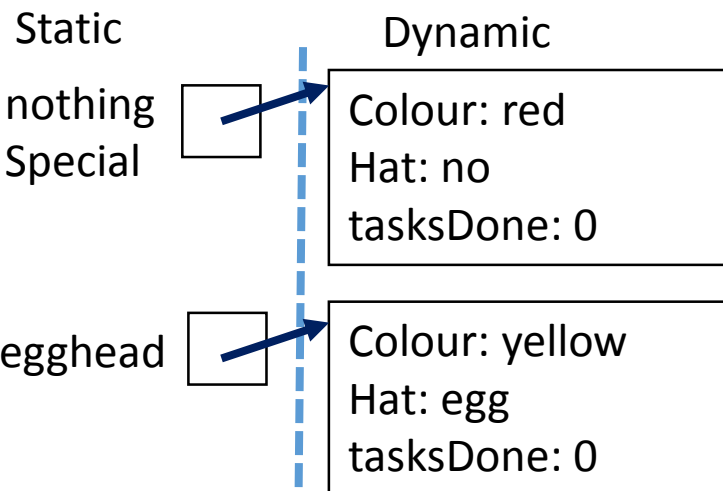
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Console

```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

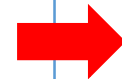
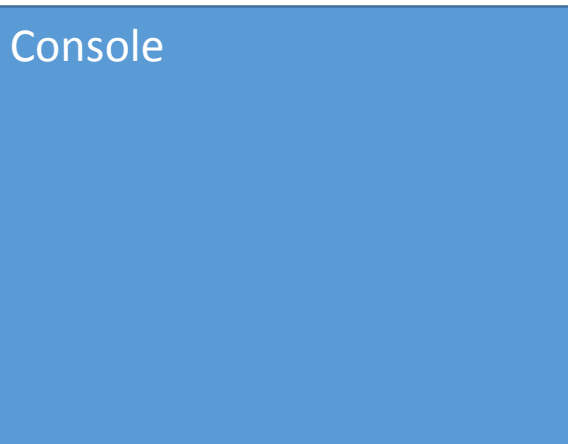
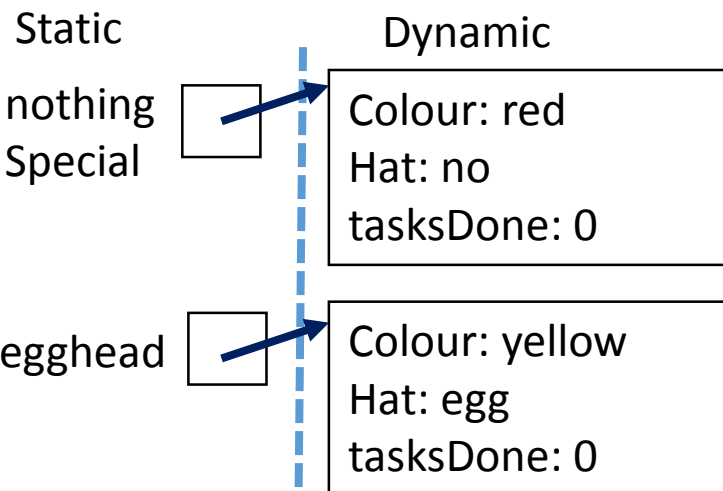
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

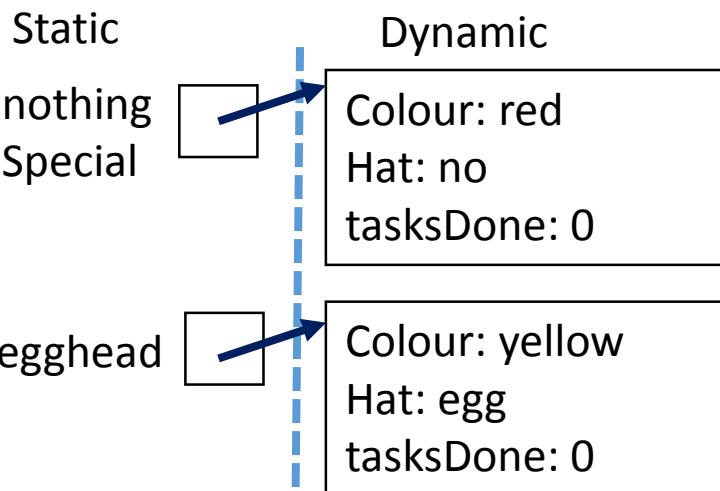
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Console
red: none hat, 0 tasks



```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

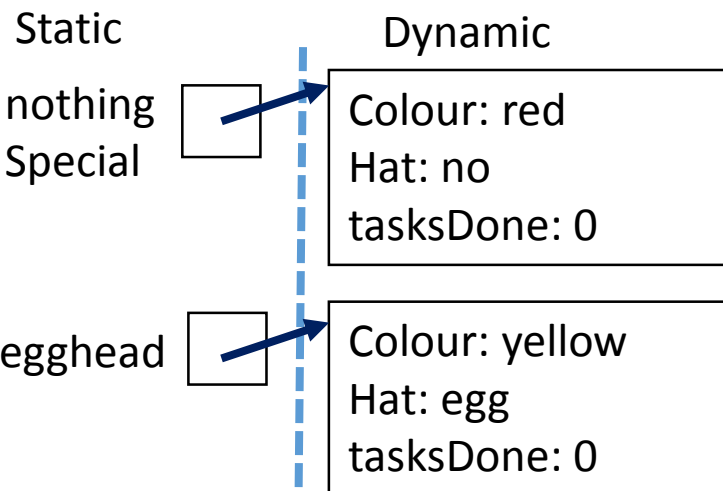
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Console
red: none hat, 0 tasks




```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1 CrewMate nothingSpecial = new CrewMate ();
2 CrewMate egghead = new CrewMate ("yellow", "egg");
3 System.out.println(nothingSpecial);
4 System.out.println(egghead);
5 egghead.completeTasks ();
6 nothingSpecial.setTasksDone (5);
7 nothingSpecial.completeTasks ();
8 System.out.println(egghead.getTasksDone ());
9 System.out.println(nothingSpecial.getTasksDone ());
10 egghead.completeTasks ();
11 System.out.println(nothingSpecial);
12 System.out.println(egghead);
    }}

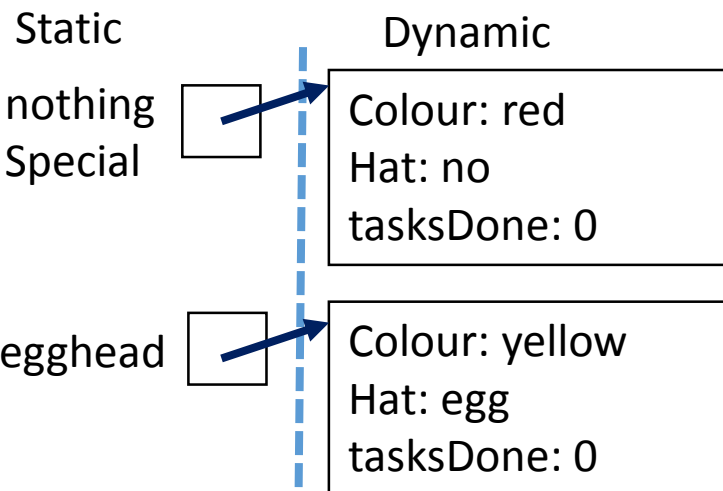
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Console
red: none hat, 0 tasks
yellow: egg hat, 0 tasks



```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1 CrewMate nothingSpecial = new CrewMate ();
2 CrewMate egghead = new CrewMate ("yellow", "egg");
3 System.out.println(nothingSpecial);
4 System.out.println(egghead);
5 egghead.completeTasks ();
6 nothingSpecial.setTasksDone (5);
7 nothingSpecial.completeTasks ();
8 System.out.println(egghead.getTasksDone ());
9 System.out.println(nothingSpecial.getTasksDone ());
10 egghead.completeTasks ();
11 System.out.println(nothingSpecial);
12 System.out.println(egghead);
    }}

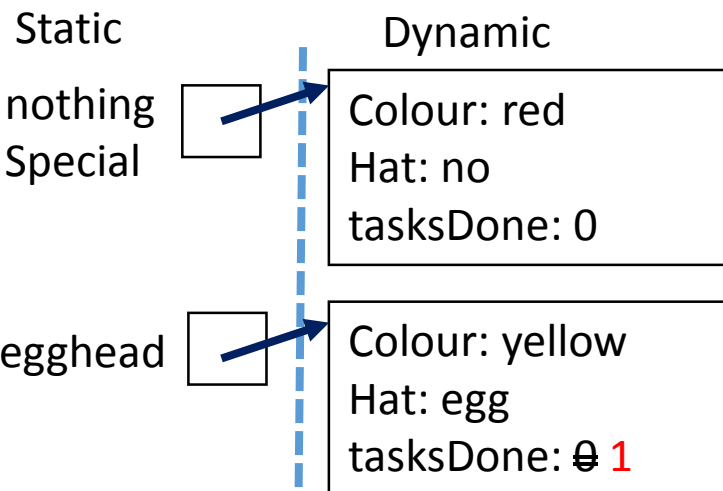
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Console
red: none hat, 0 tasks
yellow: egg hat, 0 tasks


```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println (nothingSpecial);
4   System.out.println (egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println (egghead.getTasksDone ());
9   System.out.println (nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println (nothingSpecial);
12  System.out.println (egghead);
    }
}

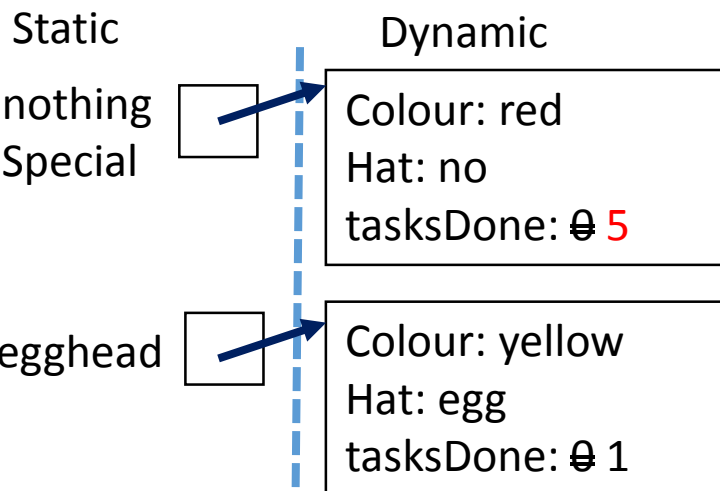
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Console
red: none hat, 0 tasks
yellow: egg hat, 0 tasks

```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }
}

```

```

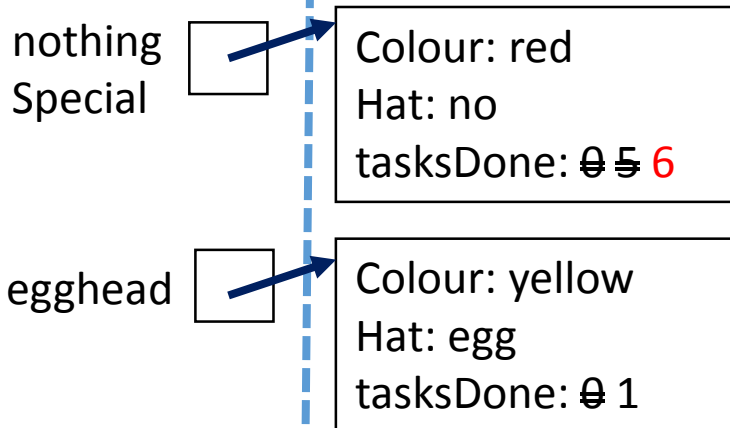
public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Static Dynamic



Console
red: none hat, 0 tasks
yellow: egg hat, 0 tasks

```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1 CrewMate nothingSpecial = new CrewMate ();
2 CrewMate egghead = new CrewMate ("yellow", "egg");
3 System.out.println(nothingSpecial);
4 System.out.println(egghead);
5 egghead.completeTasks ();
6 nothingSpecial.setTasksDone (5);
7 nothingSpecial.completeTasks ();
8 System.out.println(egghead.getTasksDone ());
9 System.out.println(nothingSpecial.getTasksDone ());
10 egghead.completeTasks ();
11 System.out.println(nothingSpecial);
12 System.out.println(egghead);
    }}

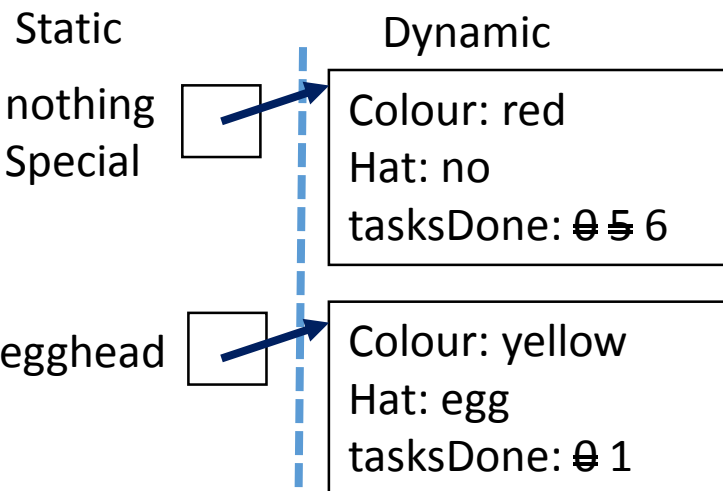
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate(String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Console
red: none hat, 0 tasks
yellow: egg hat, 0 tasks
1

```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1 CrewMate nothingSpecial = new CrewMate ();
2 CrewMate egghead = new CrewMate ("yellow", "egg");
3 System.out.println(nothingSpecial);
4 System.out.println(egghead);
5 egghead.completeTasks ();
6 nothingSpecial.setTasksDone (5);
7 nothingSpecial.completeTasks ();
8 System.out.println(egghead.getTasksDone ());
9 System.out.println(nothingSpecial.getTasksDone ());
10 egghead.completeTasks ();
11 System.out.println(nothingSpecial);
12 System.out.println(egghead);
    }}

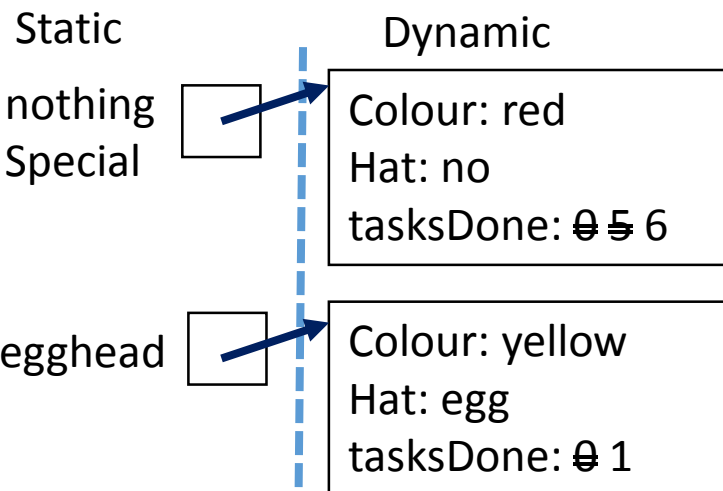
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



```

Console
red: none hat, 0 tasks
yellow: egg hat, 0 tasks
1
6

```

```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

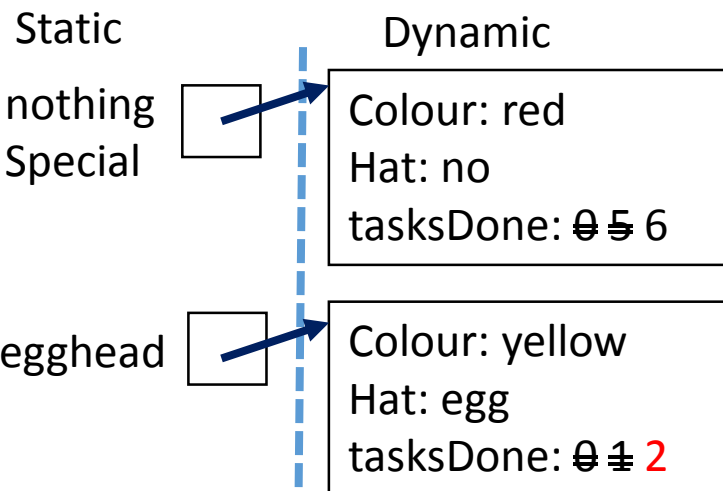
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



```

Console
red: none hat, 0 tasks
yellow: egg hat, 0 tasks
1
6

```

```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

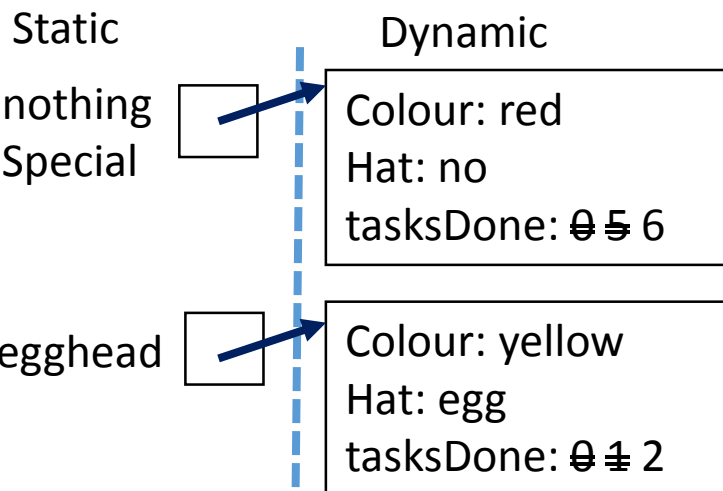
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



Console

```

red: none hat, 0 tasks
yellow: egg hat, 0 tasks
1
6
red: none hat, 6 tasks

```



```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println (nothingSpecial);
4   System.out.println (egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println (egghead.getTasksDone ());
9   System.out.println (nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println (nothingSpecial);
12  System.out.println (egghead);
    }}

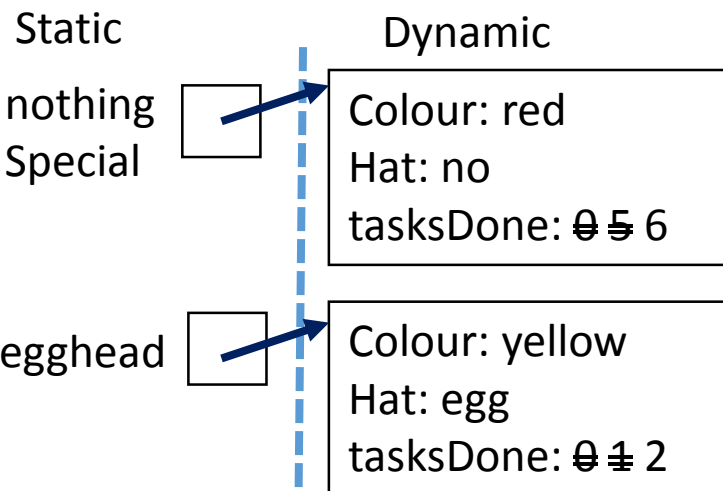
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



```

Console
red: none hat, 0 tasks
yellow: egg hat, 0 tasks
1
6
red: none hat, 6 tasks
yellow: egg hat, 2 tasks

```

```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1   CrewMate nothingSpecial = new CrewMate ();
2   CrewMate egghead = new CrewMate ("yellow", "egg");
3   System.out.println(nothingSpecial);
4   System.out.println(egghead);
5   egghead.completeTasks ();
6   nothingSpecial.setTasksDone (5);
7   nothingSpecial.completeTasks ();
8   System.out.println(egghead.getTasksDone ());
9   System.out.println(nothingSpecial.getTasksDone ());
10  egghead.completeTasks ();
11  System.out.println(nothingSpecial);
12  System.out.println(egghead);
    }}

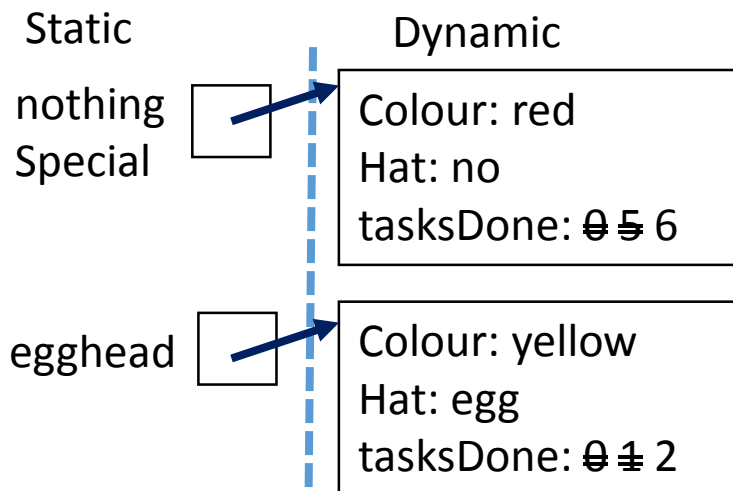
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



```

Console
red: none hat, 0 tasks
yellow: egg hat, 0 tasks
1
6
red: none hat, 6 tasks
yellow: egg hat, 2 tasks

```



```

public class AmongUs {
    public static void main(String args[]) {
        new AmongUs ();
    }
    public AmongUs () {
1 CrewMate nothingSpecial = new CrewMate ();
2 CrewMate egghead = new CrewMate ("yellow", "egg");
3 System.out.println(nothingSpecial);
4 System.out.println(egghead);
5 egghead.completeTasks ();
6 nothingSpecial.setTasksDone (5);
7 nothingSpecial.completeTasks ();
8 System.out.println(egghead.getTasksDone ());
9 System.out.println(nothingSpecial.getTasksDone ());
10 egghead.completeTasks ();
11 System.out.println(nothingSpecial);
12 System.out.println(egghead);
    }
}

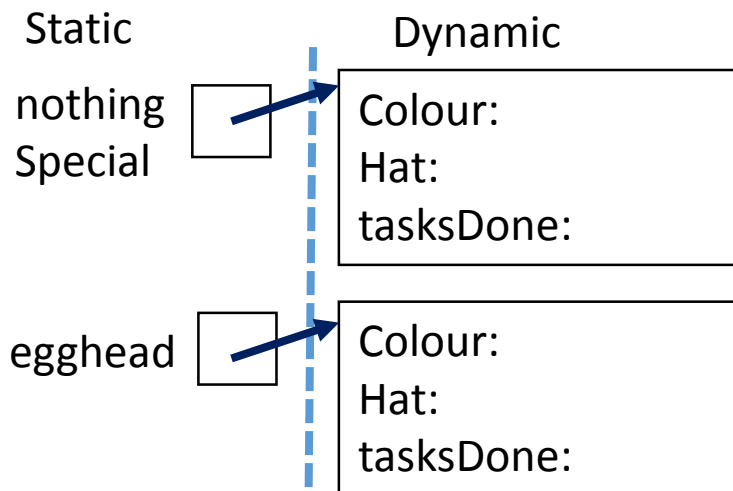
```

```

public class CrewMate {
    private String colour;
    private String hat;
    private int tasksDone;

    public CrewMate () {
        colour="red";
        hat="no";
        tasksDone=0;
    }
    public CrewMate (String c, String h) {
        colour=c;
        hat=h;
        tasksDone=0;
    }
    public int getTasksDone () {
        return tasksDone;
    }
    public void completeTasks () {
        tasksDone++;
    }
    public void setTasksDone (int n) {
        tasksDone=n;
    }
    public String toString () {
        return colour+": "+hat+" hat, " +
            tasksDone + " tasks";
    }
}

```



```

Console
red: none hat, 0 tasks
yellow: egg hat, 0 tasks
1
6
red: none hat, 6 tasks
yellow: egg hat, 2 tasks

```