Quiz 1 – Solution and Marking Scheme

Tuesday AM Solutions:

1. b, a, c, d, e, f or g or both, l, j, k, l

2.(a) 10 (b) 10 (c) 20 (d) `veloc`, `gravity`, `second` (e) `gravity` (f) `velocity`, `height` (g) n/a ** (h) `setVelocity()`, `setGravity()`, `getLaunchSpeed()` (i) Yes

(j)

```
public Rocket ( double velocity, double height ) {
this.velocity = velocity;
this.height = height;
}
```

(k) No, for many possible reasons. These include: the current class does not contain a variable called `velocity`, so `this.velocity` does not point at anything. Also, `velocity` is a private variable from another class and so cannot be accessed without using its "get" method.

** Technically all of the methods are instance methods because they do not have the word `static` in front of their names. However, `setGravity()` changes the value of a static method. Those of you with prior experience programming in Java may recognize this situation – **a method that changes a static variable should be static** so that the variable can be set without going through an instance of the class.

3.(a) line 5 (b) The values are: 10 , 4 , and 5 (c) No, because `Integer.valueOf(10)` does the same thing as autoboxing. It turns the primitive `int` 10 into an `Integer` object that points to the value 10.

Marking Scheme:

1. one mark for each answer.

2. one mark for each, except (d), (j), and (k) which were both out of two marks.

Methods can only be static or instance. So if you wrote all three methods over 2.(g) and 2.(h) without repeating yourself, you received 1/2 or 2/2. In other words, you did not lose the same mark twice in (g) and then in (h).

3.(a) one mark for the correct line and one mark for not writing an incorrect line
3.(b) one mark for formatting (not including the plus signs, but including the words, etc.) and one mark for the values.
3.(c) two marks for demonstrating understanding.

Tuesday PM Solutions:

1. a, b, c, d, h, e, l, l, j, k

2.(a) 30 (b) 10 (c) 20 (d) `veloc`, `gravity`, `second` (e) `gravity` (f) `velocity`, `height` (g) n/a ** (h) `setVelocity()`, `setGravity()`, `getLaunchSpeed()` (i) Yes

(j)

```
public Rocket ( double velocity ) {
this.velocity = velocity;
}
```

(k) No, for many possible reasons. These include: the current class does not contain a variable called `velocity`, so `this.velocity` does not point at anything. Also, `velocity` is a private variable from another class and so cannot be accessed without using its "get" method.

** Technically all of the methods are instance methods because they do not have the word `static` in front of their names. However, `setGravity()` changes the value of a static method. Those of you with prior experience programming in Java may recognize this situation – **a method that changes a static variable should be static** so that the variable can be set without going through an instance of the class.

3.(a) line 5 (b) The values are: 10 , 4 , and 5 (c) No, because `Integer.valueOf(10)` does the same thing as autoboxing. It turns the primitive `int` 10 into an `Integer` object that points to the value 10.

Marking Scheme:

1. one mark for each answer.

2. one mark for each, except (d), (j), and (k) which were both out of two marks.

Methods can only be static or instance. So if you wrote all three methods over 2.(g) and 2.(h) without repeating yourself, you received 1/2 or 2/2. In other words, you did not lose the same mark twice in (g) and then in (h).

3.(a) one mark for the correct line and one mark for not writing an incorrect line
3.(b) one mark for formatting (not including the plus signs, but including the words, etc.) and one mark for the values.
3.(c) two marks for demonstrating understanding.

Wednesday Solutions:

1. b, h, i, a, c, e, f or g or both, j, k, l

2.(a) 0 (b) 0 (c) -8 (d) `veloc`, `gravity`, `second` (e) `gravity` (f) `velocity`, `height` (g) n/a ** (h) `setVelocity()`, `setGravity()`, `getLaunchSpeed()` (i) Yes

(j)

```
public Rocket ( double velocity ) {
this.velocity = velocity;
}
```

(k) No, for many possible reasons. These include: the current class does not contain a variable called `velocity`, so `this.velocity` does not point at anything. Also, `velocity` is a private variable from another class and so cannot be accessed without using its "get" method.

** Technically all of the methods are instance methods because they do not have the word `static` in front of their names. However, `setGravity()` changes the value of a static method. Those of you with prior experience programming in Java may recognize this situation – **a method that changes a static variable should be static** so that the variable can be set without going through an instance of the class.

3.(a) line 4 (b) The values are: 10 , 4 , and 5 (c) No, because `Integer.valueOf(10)` does the same thing as autoboxing. It turns the primitive `int` 10 into an `Integer` object that points to the value 10.

Marking Scheme:

1. one mark for each answer.

2. one mark for each, except (d), (j), and (k) which were both out of two marks.

Methods can only be static or instance. So if you wrote all three methods over 2.(g) and 2.(h) without repeating yourself, you received 1/2 or 2/2. In other words, you did not lose the same mark twice in (g) and then in (h).

3.(a) one mark for the correct line and one mark for not writing an incorrect line
3.(b) one mark for formatting (not including the plus signs, but including the words, etc.) and one mark for the values.
3.(c) two marks for demonstrating understanding.

Friday Solutions:

1. b, a, c, d, i, e, f or g or both, l, j, k

2.(a) 9.8 (b) 9.8 (c) -19.6 or -9.6 (d) `veloc`, `gravity`, `second` (e) `gravity` (f) `velocity`, `height` (g) `getLaunchSpeed()` (h) `setVelocity()`, `setGravity()` (i) Yes

(j)

```
public Rocket ( double velocity, double height ) {
this.velocity = velocity;
this.height = height;
}
```

(k) No, for many possible reasons. These include: the current class does not contain a variable called `velocity`, so `this.velocity` does not point at anything. Also, `velocity` is a private variable from another class and so cannot be accessed without using its "get" method.

** Technically all of the methods are instance methods because they do not have the word `static` in front of their names. However, `setGravity()` changes the value of a static method. Those of you with prior experience programming in Java may recognize this situation – **a method that changes a static variable should be static** so that the variable can be set without going through an instance of the class.

3.(a) line 5 (b) The values are: 10 , 4 , and 5 (c) No, because `Integer.valueOf(10)` does the same thing as autoboxing. It turns the primitive `int` 10 into an `Integer` object that points to the value 10.

Marking Scheme:

1. one mark for each answer.

2. one mark for each, except (d), (j), and (k) which were both out of two marks.

Methods can only be static or instance. So if you wrote all three methods over 2.(g) and 2.(h) without repeating yourself, you received 1/2 or 2/2. In other words, you did not lose the same mark twice in (g) and then in (h).

3.(a) one mark for the correct line and one mark for not writing an incorrect line
3.(b) one mark for formatting (not including the plus signs, but including the words, etc.) and one mark for the values.
3.(c) two marks for demonstrating understanding.