Informal quiz (discuss with a partner)

1) What is the code below doing?

2) The type of `click` is ______.

3) The type of `click.getX()` is ________ and the type of `dot.getCenter()` is ________.

4) `getX()` and `getCenter()` are examples of __________. They do not take in any _________. They return ________.

5) True or False: if a graphics object is not visible on the graphics window, it does not exist.

```plaintext
click = win.getMouse()
dot = Circle(click, 20)
dot.draw(win)
```
Informal quiz (discuss with a partner)

1) What is the code below doing?
   Constructing and drawing a circle centered at the user’s click.

2) The type of click is ______.

3) The type of click.getX() is _______ and the type of dot.getCenter() is ______.

4) getX() and getCenter() are examples of __________. They do not take in any _______. They return ________.

5) True or False: if a graphics object is not visible on the graphics window, it does not exist.

   ```python
   click = win.getMouse()
   dot = Circle(click, 20)
   dot.draw(win)
   ```
Informal quiz (discuss with a partner)

1) What is the code below doing?
   Constructing and drawing a circle centered at the user’s click.

2) The type of **click** is **Point**.

3) The type of **click.getX()** is _______ and the type of **dot.getCenter()** is _______.

4) **getX()** and **getCenter()** are examples of _________. They do not take in any _______. They return _______.

5) True or False: if a graphics object is not visible on the graphics window, it does not exist.

   ```java
   Click = win.getMouse()
   dot = Circle(click, 20)
   dot.draw(win)
   ```
Informal quiz (discuss with a partner)

1) What is the code below doing?
   *Constructing and drawing a circle centered at the user’s click.*

2) The type of **click** is **Point**.

3) The type of **click.getX()** is **float** and the type of **dot.getCenter()** is **Point**.

4) **getX()** and **getCenter()** are examples of __________. They do not take in any __________. They return __________.

5) True or False: if a graphics object is not visible on the graphics window, it does not exist.

```java
click = win.getMouse()
dot = Circle(click, 20)
dot.draw(win)
```
Informal quiz (discuss with a partner)

1) What is the code below doing?
   Constructing and drawing a circle centered at the user’s click.

2) The type of **click** is **Point**.

3) The type of **click.getX()** is **float** and the type of **dot.getCenter()** is **Point**.

4) **getX()** and **getCenter()** are examples of **getters**. They do not take in any **parameters**. They return **data**.

5) True or False: if a graphics object is not visible on the graphics window, it does not exist.
Informal quiz (discuss with a partner)

1) What is the code below doing?
   Constructing and drawing a circle centered at the user’s click.

2) The type of `click` is `Point`.

3) The type of `click.getX()` is `float` and the type of `dot.getCenter()` is `Point`.

4) `getX()` and `getCenter()` are examples of *getters*. They do not take in any *parameters*. They return *data*.

5) True or False: if a graphics object is not visible on the graphics window, it does not exist.
   
   **False!**
Outline Oct 9:

- Recap user clicks and getters
- Lists as a data structure
- Mutability and modifying lists
- Functions that modify lists
- List modifying practice: build_list.py, shuffle_list.py

Notes

- Quiz 2 returned Wednesday
- Lab 4 returned Friday
- Lab 5 due Saturday night
- Office Hours 3-5pm Friday (or by appointment)
Lab 3 examples

(not posted online)
Lists
Lists as a mutable data structure

• Lists are an essential data structure, can contain basically anything (even other lists!)
Lists as a mutable data structure

• Lists are an essential data structure, can contain basically anything (even other lists!)
• Lists are mutable (we can change their data)

```python
lst1 = [5, 3, 1]
lst1[0] = 10
lst1
[10, 3, 1]
```
Lists as a mutable data structure

- Lists are an essential data structure, can contain basically anything (even other lists!)
- Lists are mutable (we can change their data)

```python
lst1 = [5, 3, 1]
lst1[0] = 10
lst1
[10, 3, 1]
```

- We can add elements to a list

```python
lst1.append(7)
lst1
[10, 3, 1, 7]
```
Lists as a mutable data structure

- Lists are an essential data structure, can contain basically anything (even other lists!)
- Lists are mutable (we can change their data)

```
lst1 = [5, 3, 1]
lst1[0] = 10
lst1
[10, 3, 1]
```

- We can add elements to a list
```
lst1.append(7)
lst1
[10, 3, 1, 7]
```

- Concatenating lists
```
lst2 = [20, 25]
lst1 + lst2
[10, 3, 1, 7, 20, 25]
```
Mutating a list changes any variables that also point to the underlying data.
List programs for today

- `build_list.py` (together)

- `shuffle_list.py` (pair programming)