Exercise: Identifying classes and methods

```
def main():
    print("Press escape to exit")
    numTargets = 2 #int(input("Enter a number of targets: "))
    win = GraphWin("Target Practice", 600, 600)
    win.setBackground("white")

    target1 = Target(Point(100,500))
    target1 .draw(win)

    target2 = Target(Point(500,500))
    target2 .draw(win)

    targets = createTargets(win, numTargets)

    numHit = 0
    key = win.checkKey()
    while key != "Escape":
        if key is not None:
           numHit += checkHitTargets(targets, key)
        moveTargets(targets)
        update(30)
        key = win.checkKey()

    print("You hit %d targets!"%numHit)

main()
```

What classes are used in main?

Exercise: Identifying classes and methods

```
def main():
    print("Press escape to exit")
    numTargets = 2 #int(input("Enter a number of targets: "))
    win = GraphWin("Target Practice", 600, 600)
    win.setBackground("white")

    target1 = Target(Point(100,500))
    target1 .draw(win)

    target2 = Target(Point(500,500))
    target2 .draw(win)

    targets = createTargets(win, numTargets)

    numHit = 0
    key = win.checkKey()
    while key != "Escape":
        if key is not None:
           numHit += checkHitTargets(targets, key)
        moveTargets(targets)
        update(30)
        key = win.checkKey()

    print("You hit %d targets!"%numHit)

main()
```

What classes are used in main?

GraphWin

Target

Point

Exercise: Identifying classes and methods

```
def main():
    print("Press escape to exit")
    numTargets = 2 #int(input("Enter a number of targets: "))
    win = GraphWin("Target Practice", 600, 600)
    win.setBackground("white")

    target1 = Target(Point(100,500))
    target1 .draw(win)

    target2 = Target(Point(500,500))
    target2 .draw(win)

    targets = createTargets(win, numTargets)

    numHit = 0
    key = win.checkKey()
    while key != "Escape":
        if key is not None:
           numHit += checkHitTargets(targets, key)
        moveTargets(targets)
        update(30)
        key = win.checkKey()

    print("You hit %d targets!"%numHit)

main()
```

What methods are called in main?

Exercise: Identifying classes and methods

```
def main():
    print("Press escape to exit")
    numTargets = 2 #int(input("Enter a number of targets: "))
    win = GraphWin("Target Practice", 600, 600)
    win.setBackground("white")

    target1 = Target(Point(100,500))
    target1.draw(win)

    target2 = Target(Point(500,500))
    target2.draw(win)

    targets = createTargets(win, numTargets)

    numHit = 0
    key = win.checkKey()
    while key != "Escape":
        if key is not None:
            numHit += checkHitTargets(targets, key)
        moveTargets(targets)
        update(30)
        key = win.checkKey()

    print("You hit %d targets!"%numHit)

main()
```

What methods are called in main?

setBackground

draw

checkKey

Exercise: Identifying classes and methods

```
def main():
    print("Press escape to exit")
    numTargets = 2 #int(input("Enter a number of targets: "))
    win = GraphWin("Target Practice", 600, 600)
    win.setBackground("white")

    target1 = Target(Point(100,500))
    target1 .draw(win)

    target2 = Target(Point(500,500))
    target2 .draw(win)

    targets = createTargets(win, numTargets)

    numHit = 0
    key = win.checkKey()
    while key != "Escape":
        if key is not None:
           numHit += checkHitTargets(targets, key)
        moveTargets(targets)
        update(30)
        key = win.checkKey()

    print("You hit %d targets!"%numHit)

main()
```

What functions are called in main?

Exercise: Identifying classes and methods

```
def main():
    print("Press escape to exit")
    numTargets = 2 #int(input("Enter a number of targets: "))
    win = GraphWin("Target Practice", 600, 600)
    win.setBackground("white")

    target1 = Target(Point(100,500))
    target1 .draw(win)

    target2 = Target(Point(500,500))
    target2 .draw(win)

    targets = createTargets(win, numTargets)

    numHit = 0
    key = win.checkKey()
    while key != "Escape":
        if key is not None:
            numHit += checkHitTargets(targets, key)
        moveTargets(targets)
        update(30)
        key = win.checkKey()

    print("You hit %d targets!"%numHit)

main()
```

What functions are called in main?

createTargets

checkHitTargets

moveTargets

update

print

input (commented out)

Exercise: Identifying classes and methods

```
def main():
    print("Press escape to exit")
    numTargets = 2 #int(input("Enter a number of targets: "))
    win = GraphWin("Target Practice", 600, 600)
    win.setBackground("white")

    target1 = Target(Point(100,500))
    target1 .draw(win)

    target2 = Target(Point(500,500))
    target2 .draw(win)

    targets = createTargets(win, numTargets)

    numHit = 0
    key = win.checkKey()
    while key != "Escape":
        if key is not None:
           numHit += checkHitTargets(targets, key)
        moveTargets(targets)
        update(30)
        key = win.checkKey()

    print("You hit %d targets!"%numHit)

main()
```

What type does the variable win have?

Exercise: Identifying classes and methods

```
def main():
    print("Press escape to exit")
    numTargets = 2 #int(input("Enter a number of targets: "))
    win = GraphWin("Target Practice", 600, 600)
    win.setBackground("white")

    target1 = Target(Point(100,500))
    target1 .draw(win)

    target2 = Target(Point(500,500))
    target2 .draw(win)

    targets = createTargets(win, numTargets)

    numHit = 0
    key = win.checkKey()
    while key != "Escape":
        if key is not None:
           numHit += checkHitTargets(targets, key)
        moveTargets(targets)
        update(30)
        key = win.checkKey()

    print("You hit %d targets!"%numHit)

main()
```

What type does the variable win have?

GraphWin

NOTE: Yes, it is an object, but its type is its class!

Exercise - method scope

```
7     def __init__(self, center):
8         """
9         Constructor. Initializes a target with the given center
10        Create three concentric circles with a character in the middle Save the
11        created shapes using member variables (perhaps a list) Compute a random
12        upwards speed for the target between 1 and 5 units per second Save the
13        speed in a member variable
14        Hint: Use random.choice(string.ascii_lowercase) to choose a random letter
15        Param self (Target): the object this function is called on
16        Param center (Point): the center of the target
17        Implicit return (Target): an instance of class Target
18        """
19        self.shapes = [Circle(center, 50), Circle(center, 40), Circle(center, 30)]
20        self.shapes[0].setFill("magenta")
21        self.shapes[1].setFill("green")
22        self.shapes[2].setFill("magenta")
23        # 1. what variables are in scope here?
24        letter = random.choice(string.ascii_lowercase)
25        self.text = Text(center, letter)
26        self.vely = 0 # TODO: Compute a random speed and save in a member variable
27        # 2. what variables are in scope here?
```

What variables are in scope on line 23?

Exercise - method scope

```
7     def __init__(self, center):
8         """
9         Constructor. Initializes a target with the given center
10        Create three concentric circles with a character in the middle Save the
11        created shapes using member variables (perhaps a list) Compute a random
12        upwards speed for the target between 1 and 5 units per second Save the
13        speed in a member variable
14        Hint: Use random.choice(string.ascii_lowercase) to choose a random letter
15        Param self (Target): the object this function is called on
16        Param center (Point): the center of the target
17        Implicit return (Target): an instance of class Target
18        """
19        self.shapes = [Circle(center, 50), Circle(center, 40), Circle(center, 30)]
20        self.shapes[0].setFill("magenta")
21        self.shapes[1].setFill("green")
22        self.shapes[2].setFill("magenta")
23        # 1. what variables are in scope here?
24        letter = random.choice(string.ascii_lowercase)
25        self.text = Text(center, letter)
26        self.vely = 0 # TODO: Compute a random speed and save in a member variable
27        # 2. what variables are in scope here?
```

What variables are in scope on line 23?

self

center

self.shapes

Exercise - method scope

```
7     def __init__(self, center):
8         """
9         Constructor. Initializes a target with the given center
10        Create three concentric circles with a character in the middle Save the
11        created shapes using member variables (perhaps a list) Compute a random
12        upwards speed for the target between 1 and 5 units per second Save the
13        speed in a member variable
14        Hint: Use random.choice(string.ascii_lowercase) to choose a random letter
15        Param self (Target): the object this function is called on
16        Param center (Point): the center of the target
17        Implicit return (Target): an instance of class Target
18        """
19        self.shapes = [Circle(center, 50), Circle(center, 40), Circle(center, 30)]
20        self.shapes[0].setFill("magenta")
21        self.shapes[1].setFill("green")
22        self.shapes[2].setFill("magenta")
23        # 1. what variables are in scope here?
24        letter = random.choice(string.ascii_lowercase)
25        self.text = Text(center, letter)
26        self.vely = 0 # TODO: Compute a random speed and save in a member variable
27        # 2. what variables are in scope here?
```

What variables are in scope on line 27?

Exercise - method scope

```
7     def __init__(self, center):
8         """
9         Constructor. Initializes a target with the given center
10        Create three concentric circles with a character in the middle Save the
11        created shapes using member variables (perhaps a list) Compute a random
12        upwards speed for the target between 1 and 5 units per second Save the
13        speed in a member variable
14        Hint: Use random.choice(string.ascii_lowercase) to choose a random letter
15        Param self (Target): the object this function is called on
16        Param center (Point): the center of the target
17        Implicit return (Target): an instance of class Target
18        """
19        self.shapes = [Circle(center, 50), Circle(center, 40), Circle(center, 30)]
20        self.shapes[0].setFill("magenta")
21        self.shapes[1].setFill("green")
22        self.shapes[2].setFill("magenta")
23        # 1. what variables are in scope here?
24        letter = random.choice(string.ascii_lowercase)
25        self.text = Text(center, letter)
26        self.vely = 0 # TODO: Compute a random speed and save in a member variable
27        # 2. what variables are in scope here?
```

What variables are in scope on line 27?

self

center

self.shapes

letter

self.text

self.vely

Exercise - method scope

```
29      def draw(self, win):
30          """
31          Calls draw(win) on the shapes of the target
32          Param self (Target): the object this function is called on
33          Param win (GraphWin): the window the draw to
34          Returns: none
35          """
36          # what variables are in scope here?
37          for shape in self.shapes:
38              shape.draw(win)
39          self.text.draw(win)
```

What variables are in scope on line 36?

Exercise - method scope

```
29      def draw(self, win):
30          """
31          Calls draw(win) on the shapes of the target
32          Param self (Target): the object this function is called on
33          Param win (GraphWin): the window the draw to
34          Returns: none
35          """
36          # what variables are in scope here?
37          for shape in self.shapes:
38              shape.draw(win)
39          self.text.draw(win)
```

What variables are in scope on line 36?

self

win

self.shapes

self.text

self.vely

Visualizing classes on the heap