```
public class Foo {

    public void fooBar() {
        moveForward(25);
    }
}
```

1. Use the class definition above to circle and identify the parts of code from the list given in parts a – j.

a) The name of the class
b) Return type of a method
c) Name of a method
d) Parameter list
e) Method call
f) Method body
g) Class definition

2. Based on this method definition, answer parts a –d.

```
public School getSchool() {

}
```

a) Which of the following is the name of the method?
   - public
   - School
   - getSchool
   - ()
   - { }

b) Which of the following is the parameter list of the method?
   - public
   - School
   - getSchool
   - ()
   - { }

c) Which of the following is the body of the method?
   - public
   - School
   - getSchool
   - ()
   - { }

d) Which of the following is the return type of the method?

School

- public
- School
- getSchool
- ()
- { }

3. Given the following method call, circle your answer for parts a – c.

```
getUpAndDance("tango.wav");
```

a) Which of the following is the argument list of the method call?

- getUpAndDance
- ("tango.wav")
- String
- ;

b) Which of the following is the name of the method being called in the method call?

- getUpAndDance
- ("tango.wav")
- String
- ;

c) Is the method that is being called inside the current class or outside the current class?

- inside
- outside

4. Given the following screenshot of Greenfoot, label the following elements:

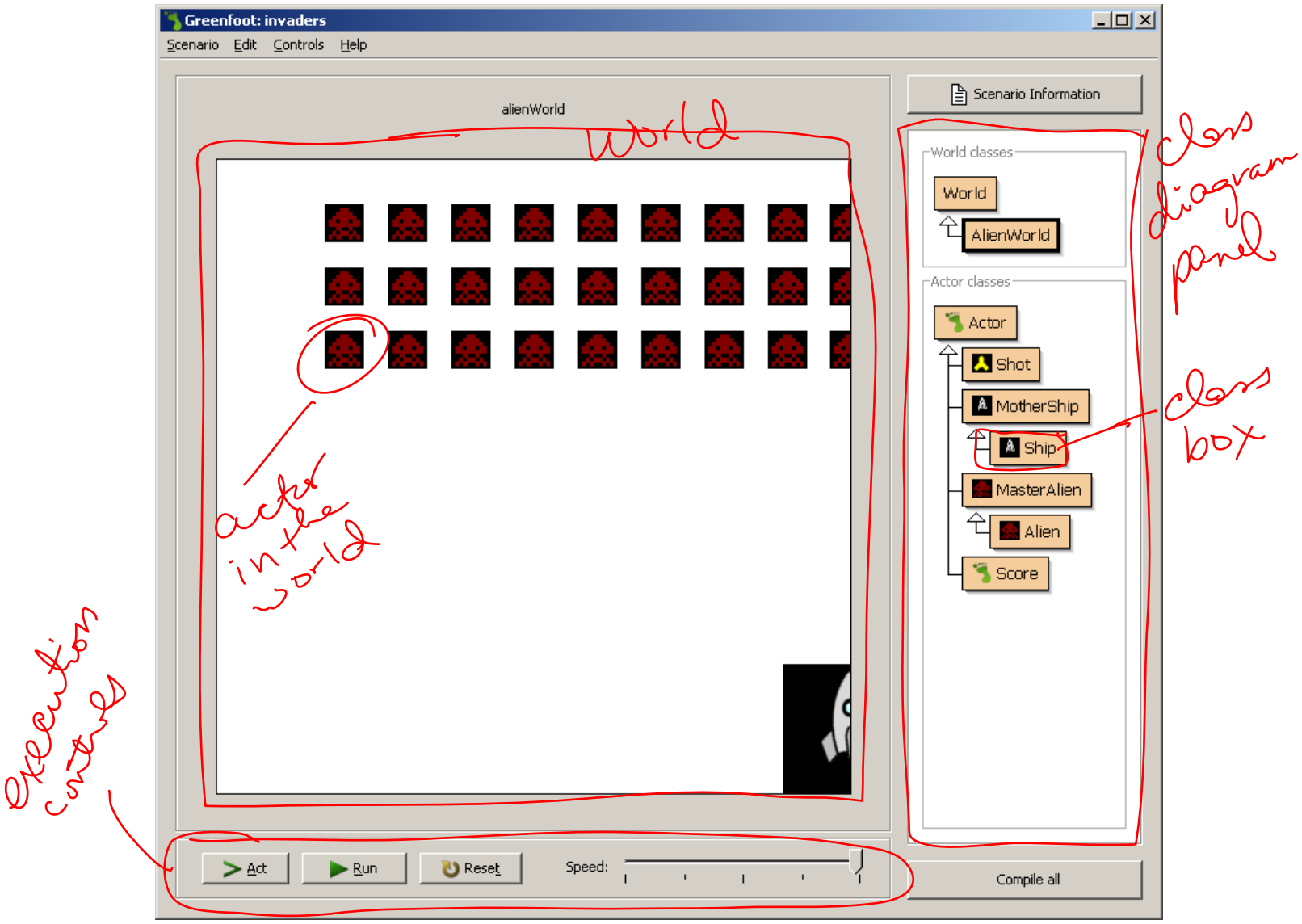
The world

An actor in the world

The class diagram panel

A class box

The execution controls



5. When would you hit the Compile All button on the Greenfoot screen? What does that button do?

You hit the compile all button when you want to compile your code. You want to compile your code when you've made changes to it and you want to translate those changes from Java source code into the language the computer can understand.

6. What is the superclass of the class Ship?

MotherShip

7. Give the name of a subclass of the class Actor.

Shot, MotherShip, MasterAlien, or Score

8. What does it mean when a method's return type is void?

The method does not return anything.

9. Why do we put comments in a Java file?

So that when others read our code, they can get an understanding of what it does. [Recall that comments are ignored by Java and are written in "English" by the programmer to be read by other programmers.]

10. Fill in the code for the following if statement so that the action given will happen 25% of the time.

```
if( Greenfoot.getRandomNumber(100) < 25 )
{
    action();
}
```

11. Fill in the ~~parameters~~ arguments to turn so that the number of degrees to turn will be between 1 and 100.

```
turn( Greenfoot.getRandomNumber(100) + 1 );
```

12. Fill in the animal class so that when the act method is called, the animal will eat food if it finds it, if it is at the world's edge, it turns 20 degrees, if it sees an enemy, it turns -50 degrees, but if it does none of these, it simply moves.

boolean atWorldsEdge() – returns true if at the edge of the world, false if not at edge.

boolean foundFood() – returns true if food is found, false otherwise

void eatFood() – eats food if food is present at the same location as the animal

void turn (int degrees) – turns the animal the specified number of degrees

boolean seeEnemy() – returns true if sees an enemy, false otherwise

void move() – moves the animal a random number of steps forward

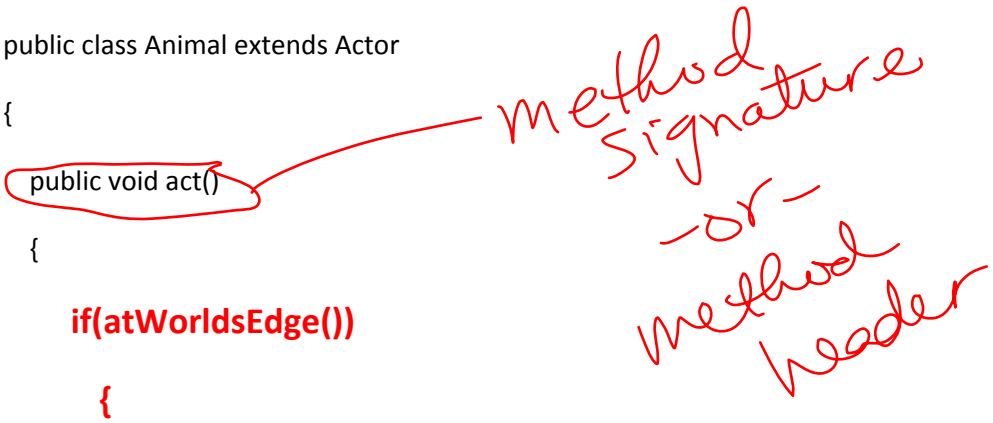
public class Animal extends Actor

```
{
   public void act()
   {
      if(foundFood())
        {
            eatFood();
        }
   }

}
public class Animal extends Actor
{
   public void act()
   {
      if(atWorldsEdge())
        {
            turn(20);
        }
   }

}

public class Animal extends Actor
{
```



```
    public void act()
    {
            if(seeEnemy())
          {
              turn(-50);
          }

    }

}

public class Animal extends Actor
{
    public void act()
    {
       move();
    }

}
```