```
public class Foo {
    private Bar  bar;
    public Foo() {
        _bar = new Bar();
    }
    public void fooBar() {
        _bar.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) Name of a constructor
f) Parameter list of a constructor
g) Method call
h) Instance variable declaration

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

3. If a class is named Can, what is the name of the class' constructor?

Can

4. Will the constructor of class Can have a return type? If Yes, what is the constructor's return type?

No

5. What is an instance variable and why do we need it?

② way for the object to store information
① Variable inside of a class

6. Fill in the agent class so that when the act method is called, if the user selects the "r" key, the agent turns a random number of degrees between 0-14 to the right (positive degrees) and if the "l" key is pressed, the agent turns a random number of degrees between 0-14 to the left (negative degrees). Assume the following methods:

void turn(int degrees) – turns the agent right the specified number of degrees.

public class Agent extends Actor

```
{
   public void act()
   {
      //Fill in your code here
   }
}
```

```
if(Greenfoot.isKeyDown("r")){
   turn(Greenfoot.randomNumber(15));
}
if(Greenfoot.isKeyDown("l")){
   turn(Greenfoot.getRandomNumber(15)-14);
}
```

7. 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, if it sees an enemy, it turns some amount of degrees, but if it does none of these, it simply moves. The animal should also keep track of how many pieces of food it has eaten.

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
{  private int foodCount;
   public void act()
   {
      if(foundFood())
      {
         eatFood();
         foodCount = foodCount +1;
      }
      if(atWorldsEdge())
      {
         turn(46);
      }
```

```
    //Fill in your code here
  }

}
```

```
if (seeEnemy())
{
  turn(15);
}
move();
```