

Refer to the Frog class to do problem 1.

```
class Frog
{
    private int location;

    public Frog()
    {
        location=0;
    }

    public void hop(int n)
    {
        location+=n;
    }

    public int getLocation()
    {
        return location;
    }

    public String toString()
    {
        String temp = "";
        for(int i = 0;i<location;i++)
            temp+=".";
        temp+=" (" +location+" )";
        return temp;
    }
}
```

1. Write an OddFrog class which extends Frog. An OddFrog's location can only ever be zero or an odd number. After hopping, an OddFrog checks if the location is even, and if it is, it adds 1 to location.

2. Write a BFrog class which extends Frog. A BFrog starts at location 100 and moves backwards instead of forwards. See sample code and output here, then write the complete BFrog class.

BFrog bff = new BFrog();	prints 100
System.out.println(bff.getLocation());	
bff.hop(20);	prints 80
System.out.println(bff.getLocation());	

3. Write code to create a regular Frog named **freddy** and then hop it 10 spaces.
4. Write code to create an OddFrog named **frankie** and hop it 11 spaces.
5. Write code to create a BFrog named **bestie** and hop it 12 spaces.
6. Assuming other code has already imported `java.util.ArrayList`, write one line of code to create an `ArrayList` named **pond** that could hold all three Frog family objects that you just created.
7. Write three lines of code to add `freddy` and `frankie` and `bestie` to the `ArrayList` you created in problem 6.
8. Given the following `Book` class, write a `.equals()` method for the `Book` class that that checks for a null object and then checks all internal values appropriately.

```
public class Book
{
    private String title;
    private String author;
    private int year;

    public Book(String t, String a, int y)
    {
        title = t;
        author = a;
        year = y;
    }
}
```