

## puzzle: Pool Puzzle



## Pool Puzzle



Your **job** is to take code snippets from the pool and place them into the blank lines in the code and output. You **may** use the same snippet more than once, and you won't need to use all the snippets. Your **goal** is to make a set of classes that will compile and run and produce the output listed.

```
_____ Nose {
```

```
_____
```

```
}
```

```
abstract class Picasso implements _____{
```

```
_____
```

```
    return 7;
```

```
}
```

```
}
```

```
class _____ { }
```

```
class _____ {
```

```
_____
```

```
    return 5;
```

```
}
```

```
}
```

```
public _____ extends Clowns {
```

```
    public static void main(String [] args) {
```

```
        _____
```

```
        i[0] = new _____
```

```
        i[1] = new _____
```

```
        i[2] = new _____
```

```
        for(int x = 0; x < 3; x++) {
```

```
            System.out.println(_____
```

```
                + " " + _____ .getClass( ) );
```

```
        }
```

```
    }
```

### Output

```
File Edit Window Help BeAfraid
```

```
% java _____
```

```
5 class Acts
```

```
7 class Clowns
```

```
_____ of76
```

Note: Each snippet from the pool can be used more than once!

```
Acts();
Nose();
Of76();
Clowns();
Picasso();
```

```
class
extends
interface
implements
i
i()
i(x)
i[x]
```

```
Of76 [] i = new Nose[3];
Of76 [ 3 ] i;
Nose [] i = new Nose();
Nose [] i = new Nose[3];
```

```
public int iMethod();
public int iMethod {}
public int iMethod () {
public int iMethod () {}
```

```
class
5 class
7 class
7 public class
```

```
i.iMethod(x)
i(x).iMethod[]
i[x].iMethod()
i[x].iMethod[]
```

```
Acts
Nose
Of76
Clowns
Picasso
```