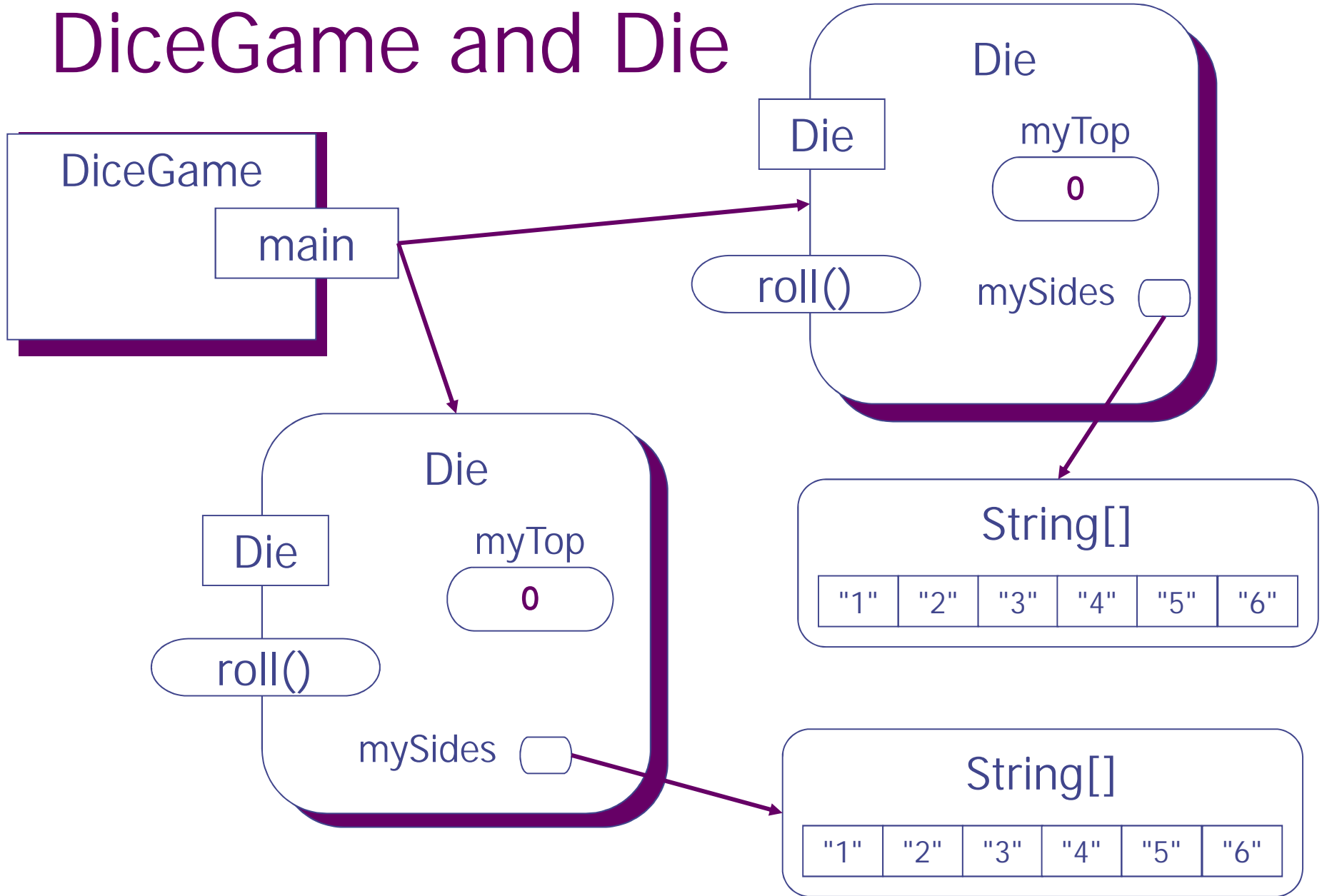


DiceGame and Die



DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1", "2", "3", "4", "5", "6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++ ) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n*" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = {"1","2","3","4","5","6"};
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

2

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++ ) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = {"1","2","3","4","5","6"};
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++ ) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

4

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++ ) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

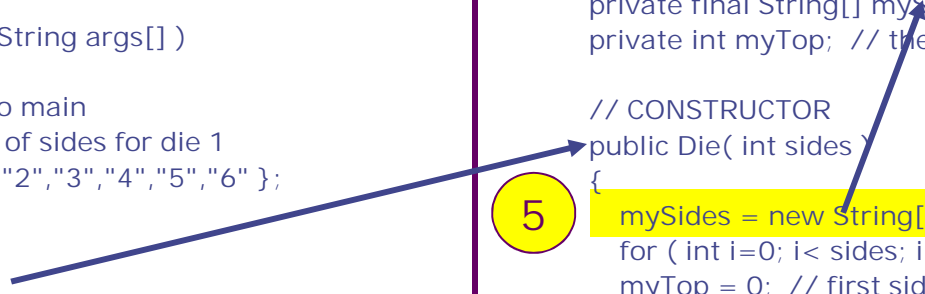
Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

5



DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

6-11

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides );

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length );
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

12

End of constructor.
Local vars/params released.
Return reference to object.

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides );

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++ ) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }

    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

13

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }

    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

14

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

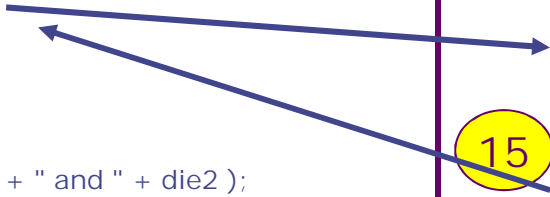
Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++ ) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

15



DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = {"1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        16 die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++ ) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

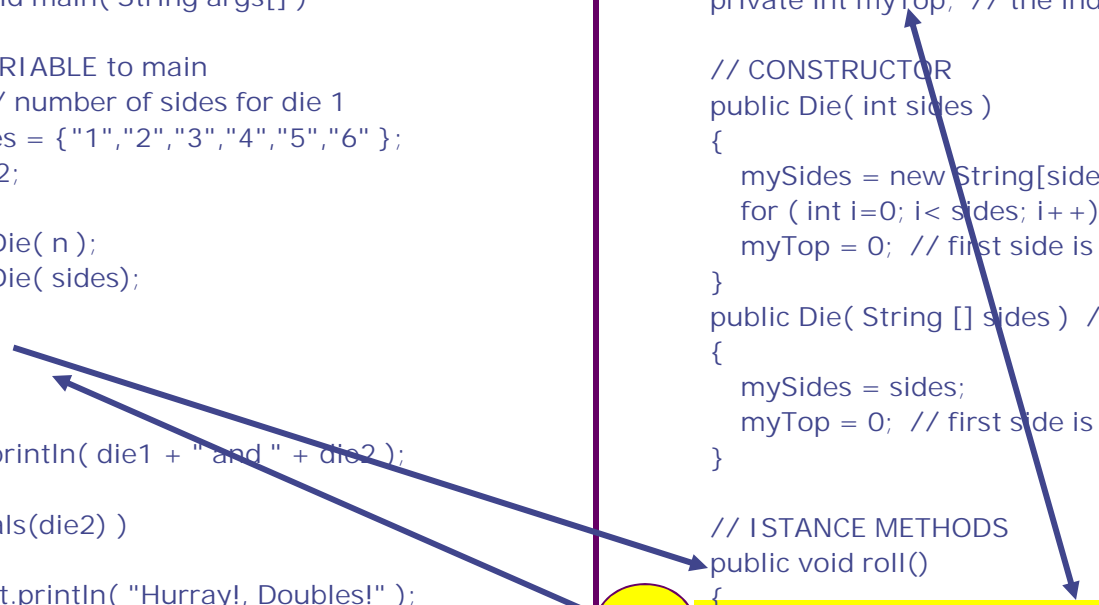
```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

17

sets and returns the contents of myTop



DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

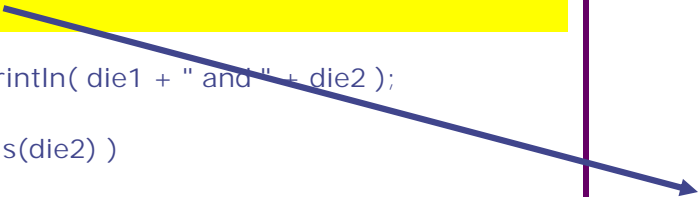
Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

18



DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

19

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = {"1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++ ) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

21

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = {"1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++ ) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

23

return mySides[myTop];

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1", "2", "3", "4", "5", "6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        24 if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1", "2", "3", "4", "5", "6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++ ) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```


DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = {"1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
END PROGRAM IS DONE EXECUTING!
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1", "2", "3", "4", "5", "6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

24a

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

25a

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1", "2", "3", "4", "5", "6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n*" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = { "1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
            "again.\n**" );
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        // LOCAL VARIABLE to main
        int n = 6; // number of sides for die 1
        String [] sides = {"1","2","3","4","5","6" };
        Die die1, die2;

        die1 = new Die( n );
        die2 = new Die( sides);

        die1.roll();
        die2.roll();

        System.out.println( die1 + " and " + die2 );

        if ( die1.equals(die2) )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

        System.out.println( "\n**\n* Please play " +
END PROGRAM IS DONE EXECUTING!
    }
}
```

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    // INSTANCE FIELDS
    private final String[] mySides; // the values for each side
    private int myTop; // the index of current value facing up

    // CONSTRUCTOR
    public Die( int sides )
    {
        mySides = new String[sides];
        for ( int i=0; i< sides; i++) { mySides[i] = ""+(i+1); }
        myTop = 0; // first side is facing up (Top)
    }
    public Die( String [] sides ) // allows die to have msgs
    {
        mySides = sides;
        myTop = 0; // first side is facing up (Top)
    }

    // INSTANCE METHODS
    public void roll()
    {
        myTop = (int) ( Math.random() * mySides.length ) ;
    }
    public String toString()
    {
        return mySides[myTop];
    }
    // Return true if same symbol is on top of this and other die
    public boolean equals( Die other ) {
        return this.toString().equals(other.toString());
    }
}
```