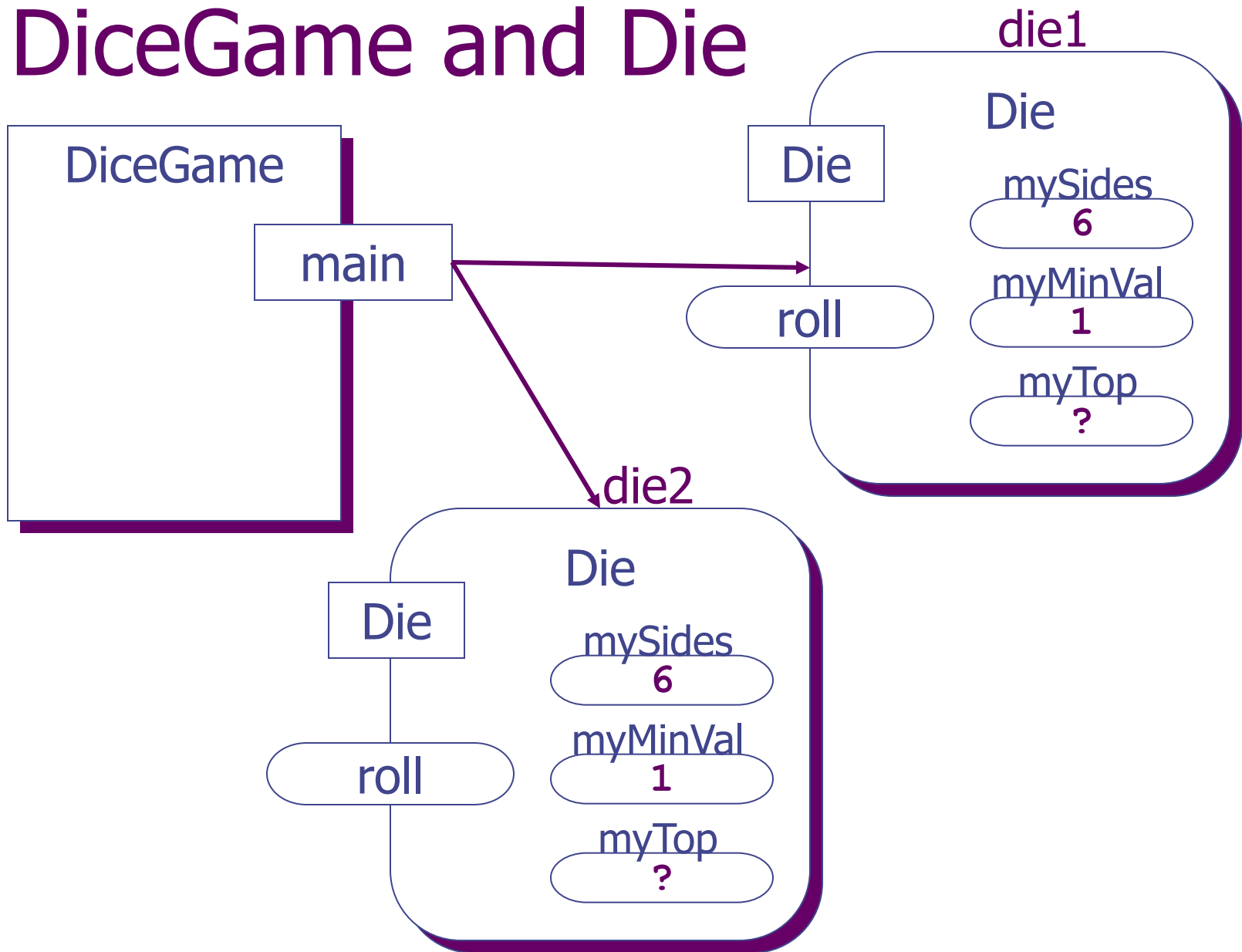


DiceGame and Die



DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
1 int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        2 Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

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

3

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        4 die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6;    // LOCAL VARIABLE to main
        Die die1, die2;  // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop;   // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        6 mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```


DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        8 myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON
    public int roll()
    {
        int roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

9

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[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

10 die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

12



DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

13

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        14 mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```


DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

15

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6;    // LOCAL VARIABLE to main
        Die die1, die2;  // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop;   // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        16 myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

17



DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        19 roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

20

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        21 int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides )
            + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```


DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

23

myTop = roll;

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        24 return myTop;
    }
}
```

returns the
contents of myTop

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

25

End of roll method.
Local variables released.
Return to caller.

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        26 roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

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

repeat
the call
to roll()
as before

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```


DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

31

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

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

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

END

PROGRAM IS DONE EXECUTING!

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            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
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```

DiceGame class

```
// MAIN CLASS for DiceGame Application
public class DiceGame
{
    // MAIN METHOD
    public static void main( String args[] )
    {
        int sides = 6; // LOCAL VARIABLE to main
        Die die1, die2; // LOCAL VARIABLES to main
        int roll1, roll2; // LOCAL VARIABLES to main

        die1 = new Die( sides, 1 );
        die2 = new Die( sides, 1 );

        roll1 = die1.roll();
        roll2 = die2.roll();

        System.out.println( roll1 + " and " + roll2 );

        if ( roll1 == roll2 )
        {
            System.out.println( "Hurray!, Doubles!" );
        }
        else
        {
            System.out.println( "Sorry, no doubles!" );
        }

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

END

PROGRAM IS DONE EXECUTING!

Die class

```
// INSTANTIABLE CLASS that represents a die
public class Die
{
    int mySides; // INSTANCE VARIABLE in a Die object
    int myMinVal; // INSTANCE VARIABLE in a Die object
    int myTop; // INSTANCE VARIABLE in a Die object

    // CONSTRUCTOR
    public Die( int sides, int min )
    {
        mySides = sides;
        myMinVal = min;
        myTop = myMinVal;
    }

    // NON-VOID INSTANCE METHOD
    public int roll()
    {
        int roll; // LOCAL VARIABLE to roll()

        roll = (int) ( Math.random() * mySides ) + myMinVal;

        myTop = roll;

        return myTop;
    }
}
```