

Unit Five Lecture Two:

Topic #1: Random Numbers

One way to create random numbers is to use the `random()` method of the `Math` class.

`Math.random()` returns a double value n , such that $0 \leq n < 1$

To effectively use `Math.random()` to create random integers, you will have to insert it into the following algorithm:

```
int x = (int)((B - A + 1) * Math.random() + A);
```

This is where you are generating random integers in the range from `A` to `B` (inclusive).

Example 1: Generate a random integer between 1 and 6, thus simulating the toss of a die.

```
int die = (int)(6 * Math.random() + 1);
```

Example 2: Generate a random integer between 50 and 100, thus simulating a test score.

```
int score = (int)(51 * Math.random() + 50);
```

Topic #2: Creating an array where each entry is unique

This is a problem that occurs frequently in computer programming. Below you will find two different ways to handle this situation.

In both of the examples, we are creating an array with 5 unique die tosses.

Brute Force Example:

```
private void setTosses()
{
    int[] tosses = new int[5];
    int num;
    boolean flag;

    for (int i=0; i<tosses.length; i++)
    {
        if (i==0)
            tosses[i] = (int)(6 * Math.random() + 1);
        else
        {
            do
            {
                num = (int)(6 * Math.random() + 1);
                flag = check(tosses, num, i);
            }
            while (!flag);

            tosses[i] = num;
        }
    }
}

private boolean check(int[] t, int n, int j)
{
    for (int k=0; k<j; k++)
    {
        if (t[k] == n) return false;
    }

    return true;
}
```

Smart Array Example:

```
private void setTosses()
{
    int[] smart = {1, 2, 3, 4, 5, 6};
    int[] tosses = new int[5];
    int num;

    for (int i=0; i<tosses.length; i++)
    {
        do
        {
            num = (int)(6 * Math.random());
        }
        while (smart[num] == 0);

        tosses[i] = smart[num];
        smart[num] = 0;
    }
}
```