Math random review

The static method named random in the Math class can be used to create a random decimal value that is greater than or equal to 0.0 and less than 1.0.

```
double x = Math.random();      // returns a double 0.0 <= x < 1.0
```

Math.ramdom can be multiplied by a constant to change the range:

```
x = Math.random() * 6;    // ______ <= x < ______
x = Math.random() * 10;   // ______ <= x < ______
```

Casting the result to an integer is used to produce whole number values: list the values below

```
num = (int) (Math.random() * 5);         // ____________________________
num = (int) (Math.random() * 2);         // ____________________________
num = (int) (Math.random() * 2) + 1;     // ____________________________
num = (int) (Math.random() * 8) + 1;     // ____________________________
num = (int) (Math.random() * 10) - 4;    // ____________________________
```

Notice the general form:

```
(int) (Math.random() * howManyNums) + startingNum;
```

Be careful to use the extra set of parentheses in the examples above since

```
num = (int) Math.random() * 6 + 1; // always equals _______
```

Complete the method to return a random whole number between low and high (inclusive).

```
public static int rollDice(int low, int high)
{

   return ____________________________________________________________
}
```

If you wanted something to occur based on a probability of say 40% you could use:

```
double x = Math.random();
if (x < 0.4)
{
     System.out.println("The event occurred.");
}
else
{
     System.out.println("The event did not occur.");
}
```

Be careful not to create a logic error when answering a question like the following. Write a method that returns "win" or "lose" each 1/2 of the time.

```
public String gameOutcome()
{
     if ((int) (Math.random() * 2) ) == 0)
     {
          return "win";
     }
     else if ((int) (Math.random() * 2) ) == 1)
     {
          return "lose";
     }
}
```