Labs 1 and 2 to be returned today, see me if you have any questions

Late policy shift, starting with Project 1: 50% reduction until 24 hours after due date

Style requirement

Academic Honesty
  • What is allowed?
  • What is not allowed?
07-1: **Academic Honesty**

- **What is allowed**: You may talk with other students about:
  - **Generic Java**: how classes work, how control structures (while/for/if/etc) work, etc.
  - Anything covered in class, in the textbook, etc.
  - **Assignment requirements**: What your code should do, what input/output should be, etc.
  - **High Level** algorithms (Example: inserting into sorted array)
07-2: Academic Honesty

• What not allowed:
  • You may not look at another student’s code
  • You may not copy another student’s code
  • You may not copy code from the web / any other source and submit it as your own work
What if we wanted random numbers (for a dice game, for example)

```java
import java.util.Random;

class Test {
    public static void main(String[] args) {
        Random r = new Random();
        int randnum = r.nextInt(25);
    }
}
```

- `r.nextInt(25)` returns a value between 0 and 24.
07-4: Random Numbers

• There are no “real” random numbers in Java
  • Computers are deterministic!
  • That is, they work the same way on the same data, every time – the output is determined by the input.

• Java uses Pseudo-Random numbers instead
  • Not “really” random, but “look” random
Pseudo-Random Numbers

- Create a sequence of numbers that “look random”
  - Bounce all over the number line
- Start with a “seed”, initial number
- Function that takes previous number, return the next one in the sequence
- \( X_i = (aX_n + c) \mod m \)
We can “seed” the Random constructor, by giving the first element in the sequence

Run the program twice, the same exact sequence of random numbers will occur, great for debugging

If we don’t give the Random constructor a seed, picks number of milliseconds that have elapsed since Jan 1, 1970
User rolls two dice up to three times to try to beat a randomly generated target.

After each roll, user decides to roll again or stick with current roll.

Driver - main

Game - play //main logic

Player - all input/output

Die - 6 sided die