# 5 Checkpoint 5

## Aim of Checkpoint

This *Checkpoint* is in two parts, the first uses random numbers to simulate "fair" dice and the second is to extend the simulation to "weighted" dice. In both cases the aim is to estimate the frequency histogram. These can be compared to the theoretical predictions from the *Probability and Statistics* parts of FOMP (or POMP), later in the year.

From a computing viewpoint this checkpoint is an introduction to the use of arrays *static methods* as well as a basic introduction to computer simulation. This checkpoint is worth **35%** of the course mark.

*It is the final checkpoint if you started at checkpoint 1.*

## Submission Dates

It is expected that this checkpoint is completed during the **fifth** or **sixth** laboratory session (if starting at checkpoint 1).

Final submission date for this checkpoint is: **5.00 pm, Thursday 3rd November**

## Random Numbers

Random numbers are used extensively in numerical simulation either as part of the algorithm or to simulate random events such as the detection of particles or the addition of noise.

Computationally random numbers are generated by an algorithm that produces a series of pseudo-random numbers, that is a deterministic series of numbers that are statistically random. A series of random numbers can be obtained by successive calls to method,

```
double Math.random();
```

which, each time it is called, returns a `double` between 0 to and 1.0 , (but not including 1.0), with a uniform probability.

## Computing Tasks

### Task 1: Statistics of Dice

**a)** Write a JAVA `int` *method* to simulate the result of rolling a $m$ sided dice which randomly gives a integer from $1 \rightarrow m$ by using `Math.random()` as a source of random numbers.

**b)** Use this *method* in a JAVA program to estimate the frequency histogram of the summation of $n$ identical dice.

Your program should use the `Display` class to prompt the user for:

- The number of sides on the "dice".
- the number of dice.
- The number of throws over which the frequency histogram should be calculated.

Your programs should then,

1. Calculate the histogram
2. Display the histogram as a graph using the `SimpleGraph` class.

Test your program with at least:

| Number of sides | Number of Dice |
|---|---|
| 6 | 1 |
| 6 | 2 |
| 6 | 8 |
| 15 | 2 |
| 15 | 8 |

at a range of different sample sizes.

**Task 2: "loaded" Dice**

**a)** Write a JAVA `int` method to simulate the result of rolling a 6 sided dice where the "dice" has been "weighted" so that the probability of rolling a **1** or a **6** is *twice* that of any other number.

Hint: Write down the probability of obtaining each number 1 to 6, do you see a pattern? Your will find the simple "fair" dice function from the previous program useful.

**b)** Modify the program from **Task 1** for the "fair" dice to estimate the frequency histogram of the sum of values when *n* of these "weighted" are thrown.

**Note:** As long as you have used a *method* in the first task then the modification to the second task in very easy.

Test your program with at least:

| Number of sides | Number of Dice |
|---|---|
| 6 | 1 |
| 6 | 2 |
| 6 | 8 |

at a range of different sample sizes.

Compare these the obtained results for those obtained for the 6 sided "fair" dice.

*Note: You may optionally combine the two programs and ask the user if they want "fair" or "unfair" dice. People who have done significant amounts of* JAVA *programming may like to use method overloading to make the code clearer.*

## End of Checkpoint

When you have completed both your program call a demonstrator and show them the code and the working output. This is the end of **checkpoint 5**. Ensure that the demonstrator checks off your name.

This is the end of the compulsory part of the course.

## Material Needed

*In addition to the material for* Checkpoint 4 *you will need material from the following documents:*

1. *Arrays*
2. *Introduction to Methods*

## What Next?

If you have started at checkpoint 1, then you have completed the compulsory part of the course. If you started at checkpoint 4, read the next section on *Introduction to Objects* and proceed to Checkpoint 6.

Additionally, if you did start on Checkpoint 1 and are going to take *Computer Simulation* next semester you are **strongly advised** to continue with the next section and attempt the final Checkpoint.