I solved exercise 1 by implementing two methods plus the main method. The first (non-main) method is used for computing the integral using the sampling method. The second method computes the integral by the hit-or-miss (short “hOm” in the code) method.
The main method is just for computing the integrals with different numbers of samples and writing the results into files. Each computation for a certain number of samples is repeated 10 times to get a value for the variance which in simply represented by the minimum and maximum value of the integral for this method and number of samples.

The graph then shows that the integral computed by both methods approaches the exact value of 0.326543 with increasing number of samples, which is no surprise. The difference between minimum (or maximum) value computed by one method and exact value, i.e. the variance, goes down roughly exponentionally.
The sampling method has the first decimal digit fixed with numberOfSamples>1, the second and third by numberOfSamples>1000. The fourth digit is never reached.
The hit-or-miss method fixes the first decimal digit at numberOfSamples>1, the second at numberOfSamples>100, the third perhaps by numberOfSamples>100000 although this is not clearly to be seen because computation then ends. Like this, sampling seems to be more accurate at the same number of steps.
Note: The number of samples can only used for comparison between these to methods and not for absolute comparisons since actually – due to runs for variance – more samples are calculated.