

```

/*CPSC assignment 1
Umar Hassan 30047693
Betty Zhang 30040611
William Chan 30041834
*/

package cpsc331.A1;

public class Hufflepuff {
    // Precondition: An integer n is given as input.
    // Postcondition: If n >= 0 then the nth Hufflepuff number, Hn, is returned as output. An
    // IllegalArgumentException is thrown otherwise.
    public static int eval(int n) {
        // Assertion: A non-negative integer n has been given as input.
        if(n >= 0) {
            if (n == 0) {
                return 10;
            } else if (n == 1) {
                return 9;
            } else if (n == 2) {
                return 8;
            } else if (n == 3) {
                return 7;
            } else {
                int hocus = 10;
                int pocus = 9;
                int abra = 8;
                int kadabra = 7;
                int i = 3;
                // Loop Invariant:
                // 1. n is an integer such that n >= 4.
                // 2. i is an integer such that 3 <= i <= n.
                // 3. Hocus is an integer variable with the value of the (i-3)th Hufflepuff
                number.
                // 4. Pocus is an integer variable with the value of the (i-2)th Hufflepuff
                number.
                // 5. Abra is an integer variable with the value of the (i-1)th Hufflepuff number.
                // 6. Kadabra is an integer variable with the value of the (i)th Hufflepuff
                number.

                // Bound Function: n-i

                while (i < n) {
                    int shazam = 4*kadabra - 6*abra + 4*pocus - hocus;
                    hocus = pocus;
                    pocus = abra;
                    abra = kadabra;
                    kadabra = shazam;
                    i += 1;
                }
                return kadabra;
            }
            // Assertion:
            // 1. A non-negative integer n has been given as input.
            // 2. The nth Hufflepuff number, Hn, has been returned as output.
        }
        else {
            throw new IllegalArgumentException("Silly muggle! The input integer cannot be negative.");
        }
        // Assertion:
        // 1. A non-negative integer n has been given as input.
        // 2. The nth Hufflepuff number, Hn, has been returned as output.
    }
    // The main method takes an integer input n as an argument in the command line.
    // The method checks if a valid argument is present in the command line, it will throw
    // an IllegalArgumentException if not. If a valid argument was given, proceed to call
    // eval function and return the corresponding Hufflepuff number to user.
    public static void main(String[] args) {

```

```

Boolean IllegalArgument=false;    //Initiate a boolean to check for illegal argument
if(args.length != 0) {    //If the argument length is not 0
    if (!args[0].matches("-?\\d+(\\.\\d+)?")) //If the first argument are numbers
    { //Set the boolean to true and throw an exception
        IllegalArgument=true;
        throw new IllegalArgumentException("Silly muggle! One integer input is
required.");
    }
} else { //If the number of arguments was 0, throw an exception
    IllegalArgument=true;
    throw new IllegalArgumentException("Silly muggle! One integer input is required.");
}
if(!IllegalArgument) { //Check the boolean and if the boolean was false then proceed
    if (args.length == 1 && args[0].matches("\\d+")) { //If the number of arguments is 1 and
the first argument are digits
        System.out.println(eval(Integer.parseInt(args[0]))); //Print the integer returned
from calling sHuffle method with first argument
    } else if (Integer.parseInt(args[0]) < 0) { //If the first argument is a negative
integer, print error
        throw new IllegalArgumentException("Silly muggle! The input integer cannot be
negative.");
    } else { //Else throw an illegal argument exception
        throw new IllegalArgumentException("Silly muggle! One integer input is
required.");
    }
}
}
}
// References:
// eval function: CPSC 331 - Assignment #1 Proving the Correctness of Simple Algorithms - and
Implementing Them as Java Programs

```