```java
package hashfunctions;

import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;

public class Driver {

    public static void main(String[] args) throws NoSuchAlgorithmException, InterruptedException {
        System.out.println("SHA-512");
        testSHA512();
        System.out.println("BCRYPT");
        testBcrypt();
    } //end method main

    static void testBcrypt() throws InterruptedException {
        long start = System.nanoTime(); //used in calculate time taken to run program

        hashPassword("Daria");
        hashPassword("password");
        hashPassword("P4ssW0rd!");

        //calculate time taken to run program
        Thread.sleep(3000);
        long duration = (System.nanoTime() - start)/1000000;
        System.out.println("TIME TAKEN: " + duration + "ms\n\n");
    } //end method testBcrypt

    static void testSHA512() throws NoSuchAlgorithmException, InterruptedException{
        long start = System.nanoTime(); //used in calculate time taken to run program

        MessageDigest md = MessageDigest.getInstance("SHA-512"); //hashing function
        SecureRandom random = new SecureRandom(); //used to generate random salt
        //specify the plaintext passwords to hash
        hashPassword(md, random, "Daria");
        hashPassword(md, random, "password");
        hashPassword(md, random, "P4ssW0rd!");

        //calculate time taken to run program
        Thread.sleep(3000);
        long duration = (System.nanoTime() - start)/1000000;
        System.out.println("TIME TAKEN: " + duration + "ms\n");
    } //end method testSHA512


    /**
     * Method to hash a given password with salt.
     * Used with BCrypt hashing function.
     * @param password - the plaintext password to be hashed
     */
    static void hashPassword(String password) {
        String salt = BCrypt.gensalt(15);
        String hashedPassword = BCrypt.hashpw(password, salt); //hash the password using password and salt
        System.out.println("Plain text password: " + password
                + "\nHash value: " + hashedPassword);
    } //end method hashPassword


    /**
     * Method to hash a given password with salt.
     * Used with SHA-512 hashing function.
     * Abstracted from https://www.javaguides.net/2020/02/java-sha-512-hash-with-salt-example.html
     * @param md - used to hash the password
     * @param random - used to generate the salt
     * @param password - the plaintext password to be hashed
     */
    static void hashPassword(MessageDigest md, SecureRandom random, String password) {
        byte[] salt = new byte[16];
        random.nextBytes(salt);
        md.update(salt); //add the salt
        //generate the hash value using digest method
        byte[] hashedPassword = md.digest(password.getBytes(StandardCharsets.UTF_8));
        StringBuilder stringBuilder = new StringBuilder();
        //for loop to convert array of bytes to String
        for(int i = 0; i<hashedPassword.length; i++) {
            stringBuilder.append(Integer.toString((hashedPassword[i] & 0xff) + 0x100, 16).substring(1));
        } //endfor
        System.out.println("Plain text password: " + password
                + "\nHash value: " +  stringBuilder.toString());
    } //end method hashPassword
} //end class Driver
```