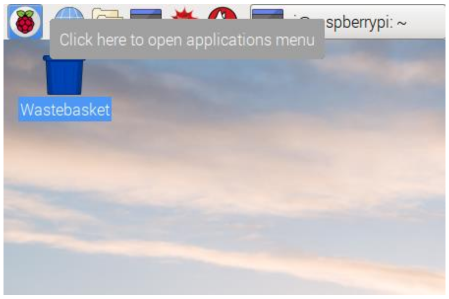
Module 2: Cybersecurity and the Raspberry Pi

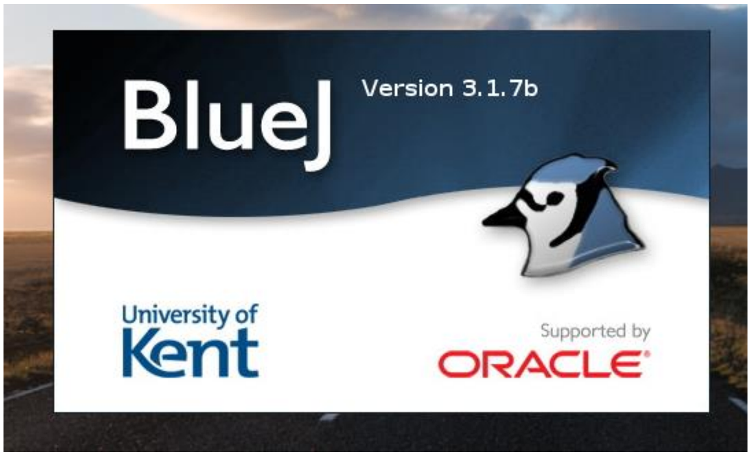
The main goal of this module is for the campers to understand the some basic concepts of cybersecurity through the Raspberry Pi. To accomplish this, the campers will learn the basic definition of cybersecurity, learn about hacking (and hackers), learn a few basic Linux commands and write and test an encryption program all on the Raspberry Pi. The campers will learn what exactly cybersecurity is and how it applies to the world. Along with this, they will learn the difference between white-hat, black-hat and grey-hat hackers. From this, the explanation of hacking will commence. The definition of hacking as well as examples of ethical and non-ethical hacking will also be described. One of the examples of hacking is penetration testing, which will be explained to the campers. This will lead into why Raspberry Pi's can be utilized for cybersecurity tasks. From here, the basic Linux commands will be demonstrated on the Raspberry Pi's. Next, the basics of encryption will be explored. Once the concept is understood, Caesar Cipher will be explained and the activity for the campers will begin. They will write a program that will utilize the Caesar Cipher and practice with encryption and cryptography on the Raspberry Pi. Once the program is written, they will add a decryption method to decrypt several encrypted messages and words. Finally, the campers will participate in a Kahoot that will recap the basics of cybersecurity, Raspberry Pi and the basics of computer science.

Encryption Activity Instructions (Module 2)

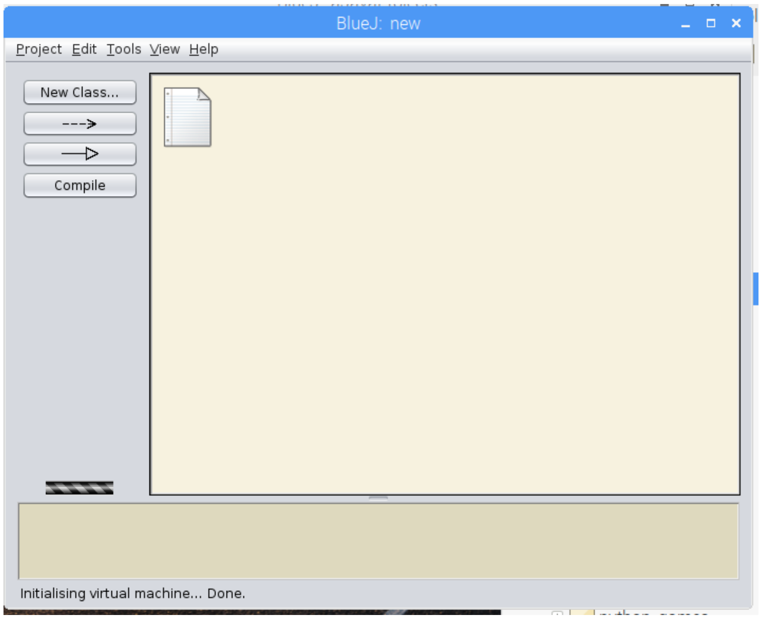
1) Navigate to the main menu and click Programming
    - You will want to find the one that is called BlueJ



2) Next you will need to execute the program. Click on it and the following window will appear



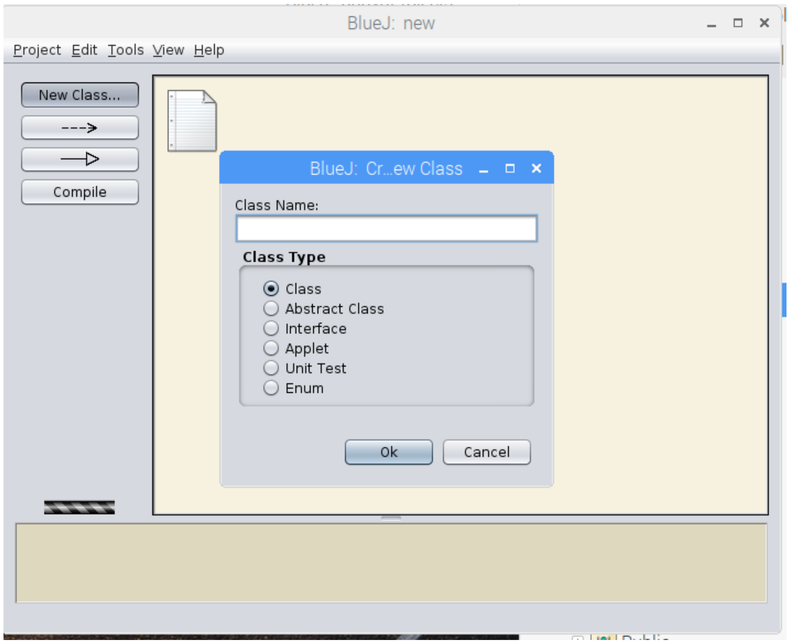
3) Next you will want to create a project. To do this click new on the top left of the window and create a folder. After you create it, the following window will appear. New will be the project name you created



Initialising virtual machine... Done.

4) Next you will need to create a class. To do this, click the new class button

- Name the class CaesarCipher and keep the class type as Class

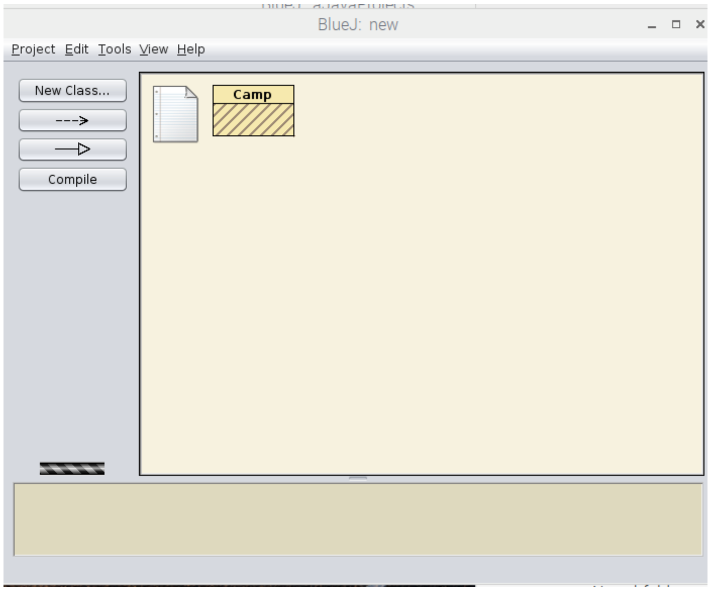


Class Name:

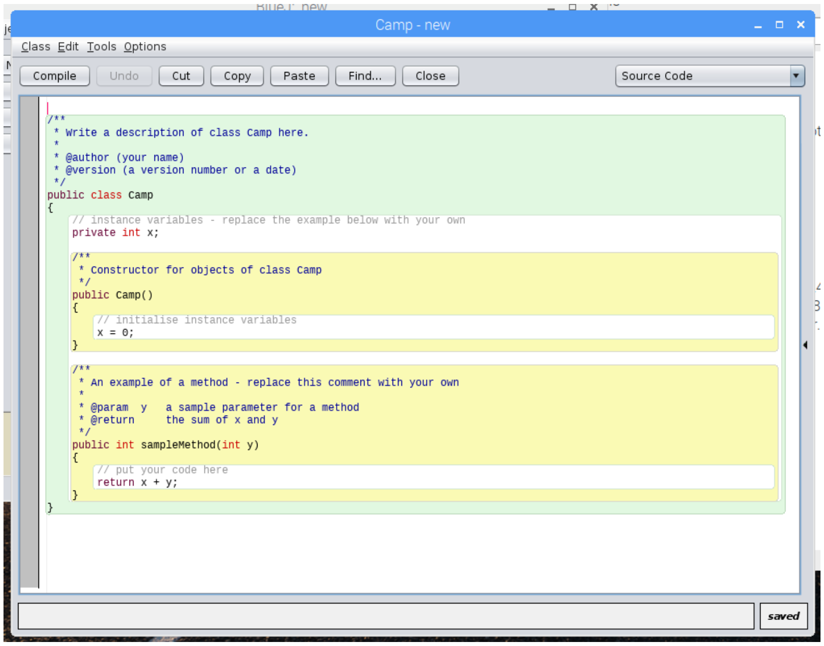
Class Type

● Class
○ Abstract Class
○ Interface
○ Applet
○ Unit Test
○ Enum

5) The following page will appear. It shall say CaesarCipher instead of Camp



6) Click on CaesarCipher and the following default code will be written



```
/**
 * Write a description of class Camp here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Camp
{
    // instance variables - replace the example below with your own
    private int x;

    /**
     * Constructor for objects of class Camp
     */
    public Camp()
    {
        // initialise instance variables
        x = 0;
    }

    /**
     * An example of a method - replace this comment with your own
     *
     * @param  y   a sample parameter for a method
     * @return     the sum of x and y
     */
    public int sampleMethod(int y)
    {
        // put your code here
        return x + y;
    }
}
```

7) The code for the cipher will be given below. NOTE: this is not the exact code I will have you write

```
import java.util.Scanner;

public class CaesarCipher {
        static String code;
        static int offset;
        static String decrypt = "ebe";

        public static String encrypt(String text, int offset) {
                StringBuffer result = new StringBuffer();
                for (int i = 0; i < text.length(); i++) {
                        if (Character.isUpperCase(text.charAt(i))) { // tests for upper case
                                char character = (char) (((int) text.charAt(i) + offset - 65) % 26 + 65);
                                result.append(character);
                        } else { // tests for lower case
                                char character = (char) (((int) text.charAt(i) + offset - 97) % 26 + 97);
                                result.append(character);
                        }
                }
                return result.toString();
        }

        public static String encode(String text, int offset) {
                StringBuilder encoded = new StringBuilder();
                for (char i : text.toCharArray()) {
                        if (Character.isLetter(i)) {
                                if (Character.isUpperCase(i)) {
                                        encoded.append((char) ('A' + (i - 'A' + offset) % 26));
                                } else {
                                        encoded.append((char) ('a' + (i - 'a' + offset) % 26));
                                }
                        } else {
                                encoded.append(i);
                        }
                }
                return encoded.toString();
        }

        public static String decrypt(String text, int offset) {
                return encrypt(text, 26 - offset);
        }

        public static String decode(String text, int offset) {
                return encode(text, 26 - offset);
        }
```

8) I will now instruct you as to how to use the code
    - To run the code you will need to compile, which is at the top left of the code screen
    - After this, open an instance of the terminal
9) From here navigate to the folder where you created the java project
    - The navigation should be cd [location]. This may be different for everyone. I will help you find the location of the file
10) To run the project, you will need the following command
    - javac [project name]
    - java [project name]
    - After the "java [project name]" line the program will run
    - This can be done in Windows machines as well