Advanced Security 1 – DT211-4

Lab Sheet 1

**Part A:**

Cryptographic applications used to lock down my web browser: Malwarebytes, MacAfee

**Applications used to see who is tracking you in your browser:**

1. Disconnect – A browser plugin that shows you which websites are tracking your browser activity in real time.
2. Privacy Badger – A browser plugin that monitors third party tracking.
3. NoScript Security Suite – also a browser plugin, supported by Edward Snowden this can be used to see who is tracking your online activity.

**Why are you being tracked in every click you make?**

Websites track every click you make in order to understand as much as possible about you, in order to serve us better but also for their own benefit.

Websites can collect lots of information from you on the web which they can then use to create revenue. They may use it to better advertise their own products to you, but they also may sell you data to other companies. They can also use the data they collect about you to measure business performance and monitor the usability of their website.

**Is it possible to prevent being tracked?**

Browsers like chrome are built around tracking the user and are virtually impossible to prevent yourself from being tracked on. However, there are alternative browsers that prevent you from being tracked online. Anonymous search engines such as StartPage, Qwant and DuckDuckGo don’t collect information. These can be combined with VPN services to help cover your tracks online. Even browsers like Tor, which hide your location and traffic still leave behind a small trail of browsing activity. To be fully anonymous you would need to boot your PC from a USB with an OS like TAILS.

**Part B:**

The deep web is called so as it is all the web pages that are not accessible by a regular search engine. They are located deeper on the web and cannot be indexed by generic search engines. However, we frequently visit the deep web when using our search engines. When we are accessing data internally on websites, we may be accessing the dark web. This is because search engines require links to display search results to us. When using a site like your local library you may search for a page that is not indexed, yet you will still be able to view this web site. Another example of when we access the deep web is when we view our personal emails. If sites like our personal emails were easily indexed by search engines, then anyone could gain access to your emails. Without the deep web we would not get as relevant search results either as there would be way more search results.

Accessing the deep web is done very easily from any regular browser as mentioned above. A way to guarantee you are on the deep web is by logging into one of your accounts. Any information you get after this point is on the deep web, as most likely it is sensitive information you would not want readily available. This is not to be confused with the dark web which requires a special browser such as Tor to be accessed. As most of us use the deep web on a daily basis there is no risk associated with using the deep web. However, the dark web is a different story. While the dark web itself is not illegal or dangerous to access, a large number of sites on the dark web are dangerous to use and illegal to access. Meanwhile the deep web is completely legal to use.

Estimates place the size of the deep web at between 96% and 99% of the internet. Meanwhile the dark web, the part of the deep web that is hidden is estimated to be anywhere from 0.1% to 5% of the internet, however nobody really knows the size of the dark web as its purpose is to be hidden.

**Part C:**

File included in submission also.

package com.lab1.calculator;  
  
/\*  
Author: Stephen Darcy  
Date: 21/09/2021  
Description: Write a simple four functions calculator (addition, subtraction, division and  
multiplication) using Java programming language (Note: you can use any programming  
language provided you will use the same for the duration of the module) which can be  
used to perform arithmetic. The calculator must have Graphical User Interface. Make  
sure that your program is well documented.  
\*/  
  
//imports  
import java.awt.event.\*;  
import javax.swing.\*;  
  
//start of calculator class  
public class Lab1 extends JFrame implements ActionListener {  
 //creating elements  
 String input1, operator, input2;  
 static JTextField *calculatorDisplay*;  
 static JFrame *frame*;  
  
 static JButton *num0*, *num1*, *num2*, *num3*, *num4*, *num5*, *num6*, *num7*, *num8*, *num9*;  
 static JButton *add*, *subtract*, *multiply*, *divide*, *equals*, *decimal*, *clear*;  
  
 //constructor  
 Lab1() {  
 input1 = "";  
 operator = "";  
 input2 = "";  
 }  
  
 //start of main  
  
 public static void main(String[] args) {  
 //creating objects  
 Lab1 cal = new Lab1();  
 *frame* = new JFrame("Calculator");  
 *calculatorDisplay* = new JTextField(15);  
 JPanel panel = new JPanel();  
  
 //adding values to the buttons  
 *num0* = new JButton("0");  
 *num1* = new JButton("1");  
 *num2* = new JButton("2");  
 *num3* = new JButton("3");  
 *num4* = new JButton("4");  
 *num5* = new JButton("5");  
 *num6* = new JButton("6");  
 *num7* = new JButton("7");  
 *num8* = new JButton("8");  
 *num9* = new JButton("9");  
 *add* = new JButton("+");  
 *subtract* = new JButton("-");  
 *multiply* = new JButton("\*");  
 *divide* = new JButton("/");  
 *equals* = new JButton("=");  
 *decimal* = new JButton(".");  
 *clear* = new JButton("Clear");  
  
 //add the buttons to the panel  
 panel.add(*calculatorDisplay*);  
 panel.add(*num1*);  
 panel.add(*num2*);  
 panel.add(*num3*);  
 panel.add(*num4*);  
 panel.add(*num5*);  
 panel.add(*num6*);  
 panel.add(*num7*);  
 panel.add(*num8*);  
 panel.add(*num9*);  
 panel.add(*num0*);  
 panel.add(*add*);  
 panel.add(*subtract*);  
 panel.add(*divide*);  
 panel.add(*multiply*);  
 panel.add(*decimal*);  
 panel.add(*equals*);  
 panel.add(*clear*);  
  
 //adding action listeners to the buttons  
 *num0*.addActionListener(cal);  
 *num1*.addActionListener(cal);  
 *num2*.addActionListener(cal);  
 *num3*.addActionListener(cal);  
 *num4*.addActionListener(cal);  
 *num5*.addActionListener(cal);  
 *num6*.addActionListener(cal);  
 *num7*.addActionListener(cal);  
 *num8*.addActionListener(cal);  
 *num9*.addActionListener(cal);  
 *add*.addActionListener(cal);  
 *subtract*.addActionListener(cal);  
 *multiply*.addActionListener(cal);  
 *divide*.addActionListener(cal);  
 *equals*.addActionListener(cal);  
 *decimal*.addActionListener(cal);  
 *clear*.addActionListener(cal);  
  
 //adding panel to the frame, setting the size, and showing  
 *frame*.add(panel);  
 *frame*.setSize(220, 250);  
 *frame*.setVisible(true);  
 }  
  
 //action listener  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 //getting input  
 String input = e.getActionCommand();  
  
 //convert to char  
 char selectedInput = input.charAt(0);  
  
 //if statement based on selected input  
 //checking to see if user selected clear  
 if (selectedInput == 'C') {  
 //clear all inputs  
 input1 = operator = input2 = "";  
 //clear screen  
 *calculatorDisplay*.setText("");  
 }  
 //checking to see if user chose a number or decimal  
 else if (selectedInput >= '0' && selectedInput <= '9' || selectedInput == '.') {  
 //check to see if operator has been chosen  
 if(!operator.equals("")) {  
 //setting as input2 if operator is already chosen  
 input2 = input2 + selectedInput;  
 }  
 else {  
 //setting as input1 if no operator has been selected.  
 input1 = input1 + selectedInput;  
 }  
 *calculatorDisplay*.setText(input1+operator+input2);  
 }  
 //check to see if user chose an operator  
 else if (selectedInput == '+' || selectedInput == '-' || selectedInput == '/' || selectedInput == '\*'){  
 //check to see if operator already selected  
 if (operator.equals("")) {  
 operator = operator + selectedInput;  
 }  
 //if an operator is already in place then you calculate the answer and reset the second input  
 else {  
 //calculate the result  
 double result = calculate(selectedInput);  
 //convert to string  
 input1 = Double.*toString*(result);  
 //set operator  
 operator = operator + selectedInput;  
 //clear second input  
 input2 = "";  
 }  
 //set the display  
 *calculatorDisplay*.setText(input1+operator+input2);  
 }  
 //final else, will only be called if operator meets no conditions, must be =  
 else {  
 //pass operator and calculate  
 double result = calculate(operator.charAt(0));  
  
 //display result  
 *calculatorDisplay*.setText(input1+operator+input2+'='+result);  
  
 //setting input1 to answer to allow the program to continue with a new second input  
 input1 = Double.*toString*(result);  
  
 //clear operator and second input  
 input2 = operator = "";  
  
 }  
 }  
  
 //method to calculate answer based on passed operator  
 private double calculate(char selectedInput) {  
 //return double based on switch  
 return switch (selectedInput) {  
 case '+' -> (Double.*parseDouble*(input1) + Double.*parseDouble*(input2));  
 case '-' -> (Double.*parseDouble*(input1) - Double.*parseDouble*(input2));  
 case '/' -> (Double.*parseDouble*(input1) / Double.*parseDouble*(input2));  
 case '\*' -> (Double.*parseDouble*(input1) \* Double.*parseDouble*(input2));  
 default -> throw new IllegalStateException("Unexpected value: " + selectedInput);  
 };  
 }  
}

**GUI:**

Graphical user interface, application

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence