

Panel 1

Last Time:

ASCII codes for characters

Type conversions : $\text{int } i = (\text{int}) c_i$ ^{char!}

String :	to Upper Case	length	index Of
	to Lower Case	charAt	<u>equals</u>
	trim	substring	

Note that `==` does not work as expected on Strings

Java API

1

Monday Quiz!

Panel 2

Program to check for Palindrome

Start with main method

Describe your algorithm via comments

For each sub task define a method

(implement method header only)

Check your logic!

→ implement each method, complex to easy

Check each method, starting with easiest.

Use plenty of Methods.

2

Panel 3

```

public class Palindrome
{
    public static String getInput()
    {
        System.out.print("Enter text: ");
        String input = Console.readString();
        return input;
    }

    public static String reverse(String input)
    {
        // take a character from end of input and
        // append to a new String; repeat
        String backwards = "";
        for (int i = input.length()-1; i >= 0; i--)
        {
            backwards += input.charAt(i);
        }
        return backwards;
    }

    public static boolean isPalindrome(String input)
    {
        // reverse input string
        String backwards = reverse(input).toUpperCase();
        // check if input equal reversed string
        if (input.equals(backwards))
            return true;
        else
            return false;
    }

    public static void main(String args[])
    {
        // Ask user to enter a String
        String input = getInput();
        // Check if it is a palindrome
        boolean answer = isPalindrome(input);
        // Provide answer
        if (answer)
            System.out.println("palindrome");
        else
            System.out.println("not a palindrome");
    }
}

```

The complete
Palindrome
class!

3

Panel 4

Could replace "isPalindrome" by a
while loop!

```

public static boolean isPalindrome(String input)
{
    // take first and last character of input
    // if they agree, take 2nd and 2nd-last and continue
    // else stop

    int i = 0;
    while ((i < input.length()) &&
           (input.charAt(i) == input.charAt(input.length()-1-i)))
    {
        i++;
    }
    if (i >= input.length())
        return true;
    else
        return false;
}

```

4

Panel 5

Coding Program:

Caesar's Cypher: *shift each letter by 1*

BERT → CFSU → BERT

↑ ↑

plain coded
or encrypted

Shift Cypher: shift each letter by x
and loop back to beginning if
necessary

BEBRA $\xrightarrow{+1}$ AFCSS $\xrightarrow{-1}$

5

Panel 6

Shift is simple to crack!

There are only 26 possible shifts.

Once you get shift, you can decode entire
message!

Substitution cypher:

↓ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ↑

↓ C D A B F E H I K J L G O M N Q R P T S Z Y X W V U ↑

26 · 25 · 24 · 23 · ...

BERT → DFPS → BERT

Total choices $26! = 403291461126605635584000000$

6

Panel 7

This code is safe ... or is it?

X YV YMMOQR
 / e e
 I or H

Most frequent letter in coded msg = most frequent letter in English
 e

?
 2nd most frequent letter ...
 most frequent pairs ...

⇒ The Code Breakers ~ WW2 - Enigma

7

Panel 8

Programming Assignment

- Program to code/decode msg via Caesar's cypher
- Program to code/decode msg via Subst. cypher
- Code Breaker Program

Due Monday (with Questions)

Final revision on Wed

8