

INTRODUCTION TO PROGRAMMING C AND JULIA

LECTURE 04: DATA TYPES; I/O STATEMENTS

Dr Ram Prasad Krishnamoorthy

*Associate Professor
School of Computing and Data Science*

ram.krish@saiuniversity.edu.in



Optimus - Tesla Robot



“eliminate dangerous, repetitive, and boring tasks”

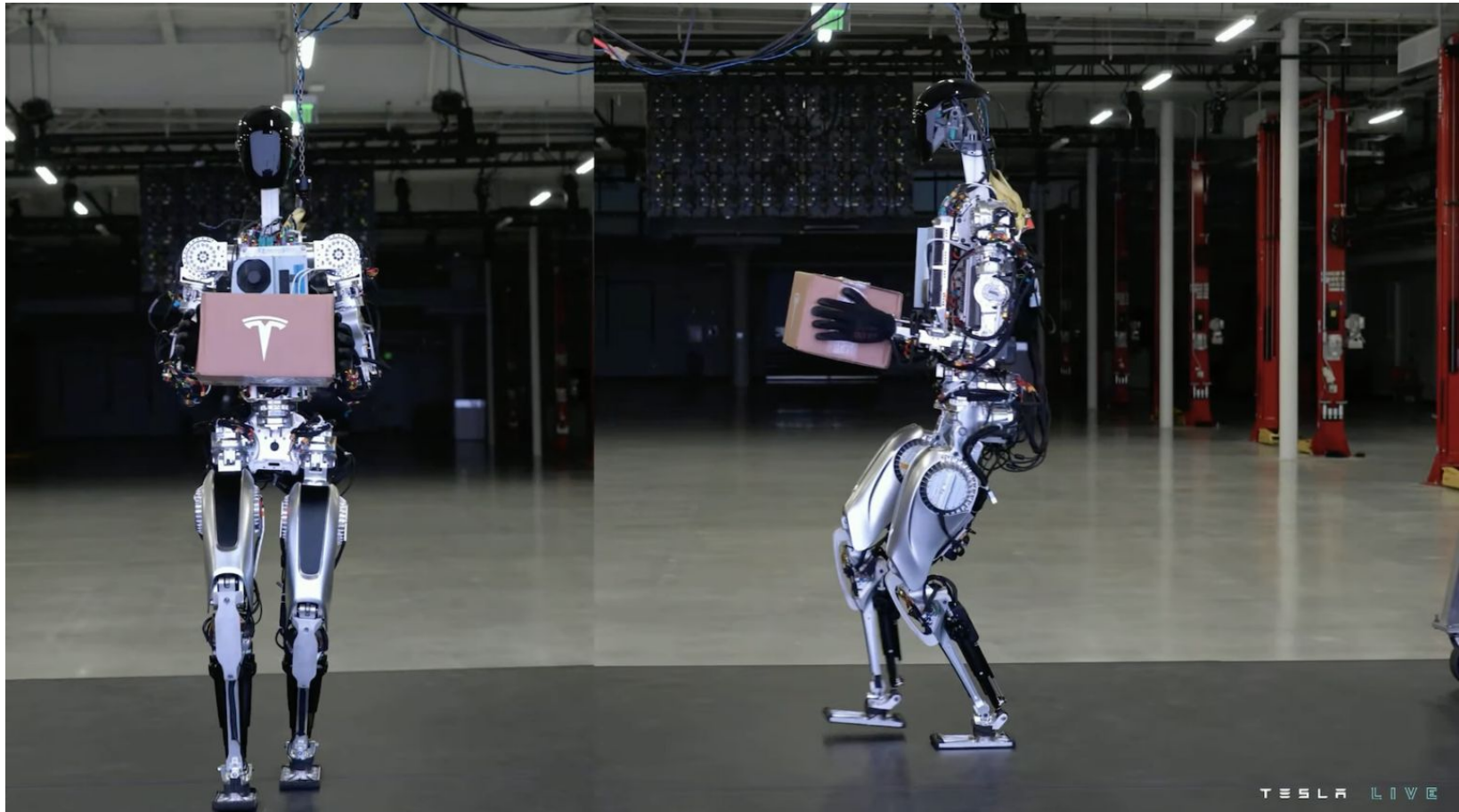
Bumble C



MISCELLANEOUS DISCUSSION



MISCELLANEOUS DISCUSSION



M I N D

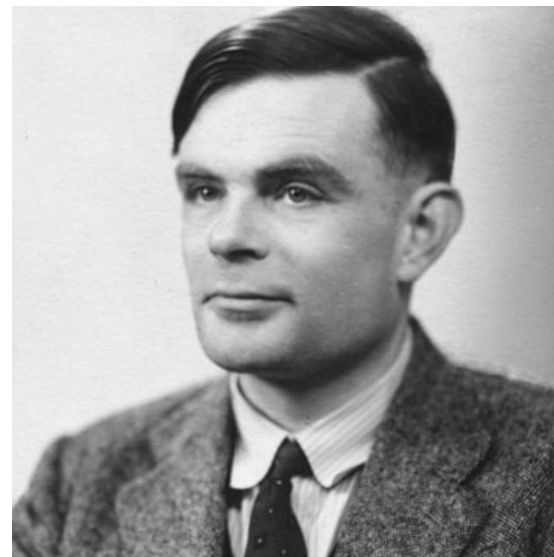
A QUARTERLY REVIEW
OF
PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

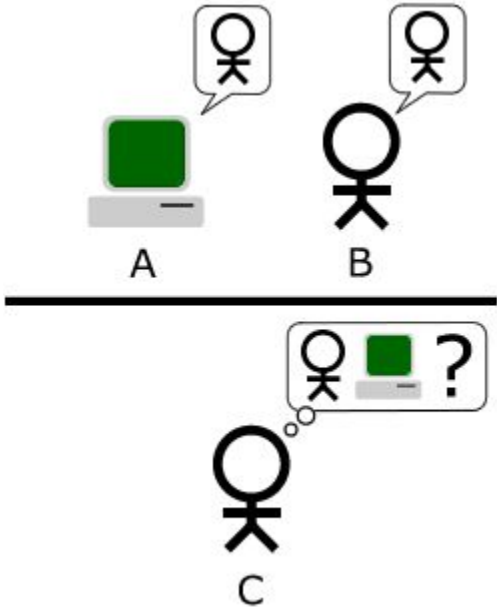
1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, 'Can machines think?' is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the



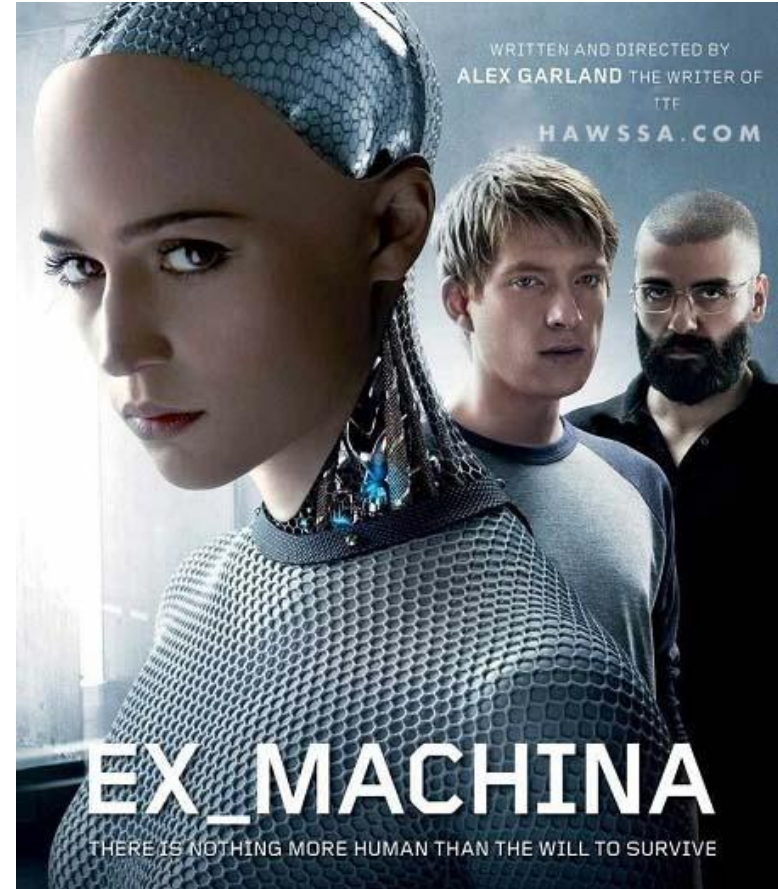
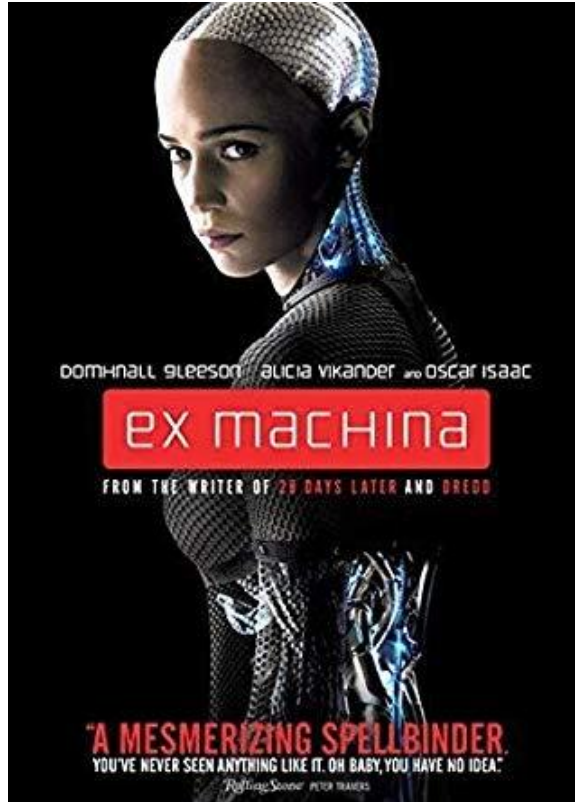
A. M. Turing, "Computing Machinery and Intelligence,"
Mind 59, no. 236 (1950): 433-460.

Turing Test



MISCELLANEOUS DISCUSSION

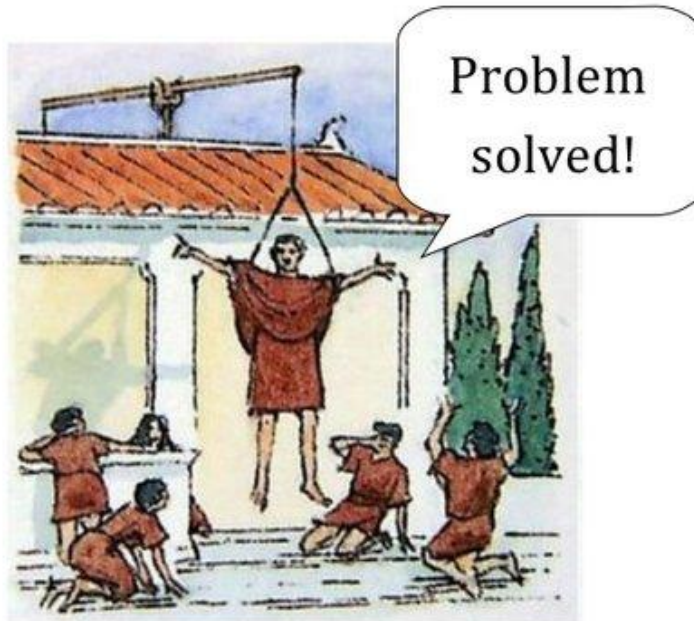
Ex Machina (2015)



MISCELLANEOUS DISCUSSION

Deus ex machina

Deus ex machina



FORMAT SPECIFIERS

FORMAT SPECIFIERS

Placeholder / Format Specifier

<i>Specifier</i>	<i>Meaning</i>	<i>Types Converted</i>
<code>%c</code>	Single character	char
<code>%d</code>	Signed decimal integer	int, short
<code>%ld</code>	Signed long decimal integer	long
<code>%f</code>	Decimal floating-point number	float, double
<code>%s</code>	Character string	char arrays
<code>%u</code>	Unsigned decimal integer	unsigned int, unsigned short
<code>%lu</code>	Unsigned long decimal integer	unsigned long

CALCULATE AVERAGE OF TWO NUMBERS

CALCULATE AVERAGE

```
#include <stdio.h>
```

```
int main(void)  
{
```

main() function with **void** as argument, this is correct as well.

```
/* Finds the average of two values and prints it. */
```

```
int value1 = 15;
```

```
int value2 = 20;
```

```
float average = 0;
```

```
average = (value1 + value2)/2;
```

```
printf("value1: %d, value2: %d, average:%f\n",  
       value1, value2, average);
```

value1: 15, value2: 20, average:17.000000

```
/* Notice the difference when divided by 2 and 2.0 */
```

```
average = (value1 + value2)/2.0;
```

```
printf("value1: %d, value2: %d, average:%f\n",  
       value1, value2, average);
```

value1: 15, value2: 20, average:17.500000

```
return 0;
```

```
}
```


PRINT TO STDOUT/TERMINAL

`printf()`

PRINT TO STDOUT

printf()

General syntax:

```
printf(format-string[,arguments,...]);
```

Example:

```
printf("Hello World\n");  
printf("Average is %f\n", average);
```

Note:

`#include <stdio.h>` is mandatory

READ FROM TERMINAL

`scanf()`

READ FROM USER

scanf()

General syntax:

```
scanf(format-string[,arguments,...]);
```

Example:

```
int x, y;  
scanf("%d %d", &x, &y);  
printf("x: %d, y: %d", x, y);
```

Note:

`#include <stdio.h>` is mandatory

`&` - address of operator is important (*pointers topics*)

READ FROM USER

```
#include <stdio.h>

int main(void)
{
    /* Finds the average of two values and prints it. */

    int value1 = 0;
    int value2 = 0;

    float average = 0;

    printf("Enter value1: ");
    scanf("%d", &value1);

    printf("Enter value2: ");
    scanf("%d", &value2);

    average = (value1 + value2)/2.0;

    printf("value1: %d, value2: %d, average:%f\n",
           value1, value2, average);

    return 0;
}
```

Enter value1: 15
Enter value2: 20
value1: 15, value2: 20, average:17.500000

Note:

- When using the `scanf()` statement, always remember to precede it with a `printf()` statement to indicate to the end-user what data should be entered.
- It will be also more appropriate if the datatype is also mentioned.
- The documentation in terms of comments is important for the readability of the program, but appropriate `printf()` statements will be helpful in running the program without errors when data is entered by user during the execution of the program.