

Exercise 7: Programming Languages

Model 1 Low-Level Languages

The following program, shown in three different languages, calculates the sum of numbers from 1 to 10. In other words, it adds $1 + 2 + \dots + 10 = 55$.

Machine Code (1st Generation)	Y86-64 Assembly (2nd Generation)	Standard C (3rd Generation)
0x000: 0x000: 70000100000000000000	.pos 0 code jmp _start	
0x100: 0x100: 0x100: 30f00b00000000000000 0x10a: 30f30100000000000000 0x114: 30f10200000000000000 0x11e: 30f20100000000000000	.pos 0x100 code _start: irmovq \$0xb, %rax irmovq \$0x1, %rbx irmovq \$0x2, %rcx irmovq \$0x1, %rdx	int main() { int upper = 11; int sum = 1; int val = 2;
0x128: 2017 0x12a: 6107 0x12c: 73460100000000000000	rrmovq %rcx, %rdi subq %rax, %rdi je done	while (val < upper) { sum = sum + val; val++; }
0x135: 0x135: 6013 0x137: 6021	loop: addq %rcx, %rbx addq %rdx, %rcx	} }
0x139: 2017 0x13b: 6107 0x13d: 74350100000000000000	rrmovq %rcx, %rdi subq %rax, %rdi jne loop	
0x146: 0x146: 00	done: halt	

1. Compare the length of each program. Do not count labels (e.g. 0x000:, .pos 0 code) or punctuation (e.g., {, }).

a) How many instructions of machine code?

b) How many instructions of assembly code?

c) How many non-blank, non-brace lines of C code?

2. The instruction `irmovq` means “move immediate value to register”. Immediate values begin with a dollar sign (\$), and registers begin with a percent sign (%).

- a) What is the value 11 in assembly code?
- b) Does assembly use decimal or hexadecimal?
- c) Does Standard C use decimal or hexadecimal?

3. In terms of a computer and CPU, what does an assignment statement do? As part of your answer, name the instructions in Model 1 that perform assignment.

4. The instruction `je` means “jump if the last operation’s result equals 0”, and the instruction `jne` means “jump if the last operation’s result does not equal 0”. Circle the portion of assembly code that corresponds to the `while` loop in C.

Model 2 High-Level Languages

In addition to adding the numbers from 1 to 10, this program prints (displays) the result on the screen using Standard I/O.

Standard C (3rd Generation)

```
#include <stdio.h>

int main()
{
    int upper = 11;
    int sum = 1;
    int val = 2;

    while (val < upper)
    {
        sum = sum + val;
        val++;
    }

    printf("Sum = %d\n", sum);
}
```

Python (4th Generation)

```
upper = 11
isum = 1
val = 2

while val < upper:
    isum = isum + val
    val = val + 1

print("Sum = " + str(isum))
```

5. Compare the C code with the Python code.

- Circle the lines of C code that were not present in Model 1.
- Which lines of C are not present (i.e., needed) in Python?
- What punctuation used in C is not required in Python?

6. Without using braces, how does Python know which lines are part of the `while` loop?

7. Why does Python use the name `isum` instead of `sum`? Hint: type `sum` into a Python shell.

8. In Python, the `range` function can be used to generate a sequence of numbers. Use a Python shell to answer this question.

- a) What is the result of `list(range(5))`?
- b) What is the result of `str(range(5))`?
- c) What do the `list` and `str` functions do?
- d) What is the result of `sum(range(5))`?
- e) What does the `sum` function do?

9. Rewrite the entire program of Model 2 using one line of Python code. Hint: you'll need to use `print`, `str`, `sum`, and `range`.

10. Based on Model 1 and Model 2, what does it mean to be low-level vs high-level?

Model 3 Guessing Game

Create a new file named `guess.py` in Thonny and enter the following code. Replace the name in Line 2 with your own name. Be careful to type the code *exactly* as shown.

```
1 name = input("What is your name? ")
2 if name == "Sharon":
3     print (name, " is a great name!")
4 else:
5     print (name, "is an okay name.")
```

Note: `input` is a **function** that displays a **prompt** on the screen and reads a line from the keyboard. In this program, the result of `input` is stored in the **variable** `name`.

11. What is the prompt? Why is there a space at the end of it?

12. Run the program a few times, entering a different name each time. Feel free to modify the messages as you see fit.

13. Now modify your Python program in Thonny as given below (change the name to your name). The library **random** provides the ability to create a random number. Read the comments and after reading the comments on lines 21 - 26 and comment on line 30, complete the program to accomplish what the comments direct you to do. Once you have programmed and tested in Thonny, copy and paste your answer below

```
1 import random
2
3 name = input("What is your name? ")
4 if name == "Sharon":
5     print (name, " is a great name!")
6 else:
7     print (name, " is an okay name.")
8 print
9
10 like = input(name + ", would you like to play guess my number? ")
11 if like == "yes":
12     print ("Great, let's play!")
13 else:
14     print ("Too bad, we're going to play anyway")
15 print()
16
17 # random number between 1 and 100
18 number = random.randrange(1, 100)
```

```
19 guess = -1
20
21 #complete the code to loop until the guessed
22 #number equals the random number generated.
23 #If the number guessed is too low or too high,
24 #your program needs to let the user know if too high or too low.
25 #Remember Activity 1 Unit 1?
26 #Hint: a while loop is needed
27
28 guess = int(input("Guess my number: "))
29
30 # check if answer is correct, too low or too high
31
32 print ("You got it!")    #when the correct guess is made
```

14. What is the difference between = and == in the programs you have written today?