

# Programming Knowledge & Skills

## Example Solutions

### 1. Basic Data Types

- a. String or Char
- b. Integer Numbers
- c. Floating Point and/or Fixed Point Numbers

*1pt scored for each correct data type listed*

*Total → 3 points*

---

### 2. What are some limitations of the different number data types?

Answers should include - but are not limited to - some of the following ideas:

- Integer number sizes are limited by the amount of memory that can be allocated to it. For example, an 8-bit unsigned integer is limited to the numbers between 0 and 255
- Trying to store a number outside the range of the allowed integer type results in an *overflow* and can cause errors
- Integers can be either signed or unsigned, and which they are affects the range of allowed numbers (and 8-bit signed integer is limited to -128 to 127)
- Floating-point numbers have a limited precision, and are only an approximation of the decimal number they represent
- Doing arithmetic with floating-point numbers can result in errors, such as  $0.2 + 0.1 = 0.300000000000000004$
- Floating-point numbers are also limited in size based on allowed memory allocation (with a common limit being  $3.4028235 \times 10^{38}$ )
- Other correct answers can be accepted at the discretion of the assessor.

*2pts for identifying a limitation of integer number types*

*2pts for identifying a limitation of floating-point number types*

*Total → 4 points*

**For the following questions, links have been provided to external, online resources which contain examples of how to perform various actions in the syntax of a large number of different programming languages.**

A discussion of possible ways of administering and evaluating the technical section of the assessment is included in the Rationale section of the assessment document itself. Most importantly, answers must be assessed on:

- 1) Does the program or answer work in the LANGUAGE OF CHOICE?
  - 2) Does the program work for the standard-use case provided in the question?
- 

3. In <LANGUAGE OF CHOICE>, perform the following conversions/type casts:
  - a. Integer to String
  - b. String to Integer
  - c. Integer to Float
  - d. Float to String

[Type-Casting Syntax in Many Languages](#)

*1pt for every correct type-cast*

*Total → 4 points*

---

4. Fibonacci Numbers Examples
  - a. [Fibonacci Sequence Examples in Many Languages](#)
5. Custom zip() Function Examples
  - a. [Appendix A: Custom Zip\(\) Functions](#)
6. Custom sort() Function Examples
  - a. [Sort Function Examples in a Variety of Languages](#)
7. Importing Functions & Concerns
  - a. [Import Syntax in Most Languages](#)

*1pt for a function that executes with no errors*

*3pts for a function that executes the performed function in the standard-use case*

*Total → 16pts for all coding tasks*

**TOTAL OVERALL: 37 points**

## Appendix A: Custom Zip() Functions (WORK IN PROGRESS)

Python:

```
def zip(list1, list2):
    result = []
    shorterList = list1 if len(list1) < len(list2) else list2
    for i in range(len(shorterList)):
        result.append([list1[i],list2[i]])

    return result
```

JavaScript:

```
function zip(list1, list2) {
    result = [];
    shorterList = list1.length < list2.length ? list1 : list2;

    for (let i=0; i< shorterList.length; i++) {
        result.push([list1[i],list2[i]])
    }
    return result;
}
```

C++:

[StackOverflow Solution \(to be tested\)](#)

Java:

[StackOverflow Solution \(to be tested\)](#)

[Solution Using Non-Standard Libraries](#)