

## A2: Design Issues of Language's Basic Components

**Note:** Here “your opinion” refers to “your group’s consensus”, not each individual member’s or speaker’s opinion, i.e. “you” refer to “your group”, not individual member of a group.

**Question 1: Names and Conventions**

- (1) Identifier rules: what are your language’s rules for identifiers? Provide 2 examples of valid identifiers and two examples of invalid identifiers. [Can you find some weird but valid identifiers?] optional
- (2) Does your language’s documentation provide guidelines for naming variables (e.g. camelCase or snake)? Does the language’s compiler/interpreter enforce such guidelines? Do you follow these guidelines in coding? Why or why not?

**Question 2: Data types**

- (1) Convert the following Java code into that of your language. Run and display the results.
- (2) Use the converted code to explain: Is type declaration required in your language? How does your language catch type errors (e.g. during compilation/interpretation or during execution)? Does your language support a Boolean type? How does your language represent a Boolean value (e.g. true or false, t or f, ...) and a Boolean type (if applicable)?

```
int x = 5, y = 10;

// Comparison operators
boolean isGreater = x > y;
boolean isNotEqual = x != y;

// Logical operators
boolean isTrue = true;
boolean isFalse = false;
boolean andResult = isTrue && isFalse;
boolean orResult = isTrue || isFalse;
boolean notResult = !isTrue;

// Compound expressions
boolean complexExpression = (x > 0) && (y < 20) || (x == 5);
```

**Question 3. Map/Dictionary Types**

Does your language have a (built-in) map/dictionary like data type? If yes, what is the name of the type? Give an example (write 1 or 2 lines of code) to show how to define and use such a type. If not, what alternatives can be used to represent map/dictionary like data type? Show an example to illustrate.

#### Question 4. Array types.

- (1) For arrays or array-like data type(s) supported by your language, where will the arrays be allocated (i.e. stack, heap, or either)? And, what categories of arrays (static, fixed stack dynamic, fixed heap dynamic, heap dynamic) the language support? For each category supported, write a line of code to give an example.
- (2) Can arrays (elements) be heterogeneous or must be homogeneous?
- (3) Can array subscripts be negative (i.e. data[-5]) in your language?
- (4) How to define and use a two-dimensional array in your language? Show 1 or 2 lines of codes that illustrate the definition and usage (i.e. accessing elements) of a 2-D array.

#### Question 5. Scoping

Some languages use nested scopes for function definitions (i.e. one function could be defined inside another function) and some languages also introduce keywords such as global, non-local etc. to indicate variable scopes.

- (1) Discuss whether your language allow nested scope for function definitions or not. If yes, give an example. If not, briefly describe the advantage(s) of not supporting the nested scope.
- (2) Discuss how your language define/distinguish global and local variables and whether keywords such as global, non-local etc. exist to describe/define variable scopes.

**Note:** if some of questions doesn't apply to your language, please say so and then pick up a feature of your language and discuss it as substitute to that question.

#### Requirements:

- 1) Create a slides file that includes answers to each of above questions;
- 2) Selected groups will present at a class meeting (date of sharing/presentation is given on Canvas) while non-presenting groups will be in charge of Q&A section.
- 3) Submit the slides (in .pptx or .pdf format) on Canvas after the presentation (by the end of the day.)

#### Grading:

- (1) Presentation or Q&As: 10 points. Presentation: clarity in content delivery, constructive code examples, and effective response to questions. Q&As: at least 2 question are required; quality of the questions.
- (2) Slides: 3 points – quality of slides, timely submission.
- (3) Evaluation of other presentations: 4 points – thoughtful take aways and constructive feedback. Note: everyone in the group needs to evaluate 1 or 2 other groups. The score here represents the average of group members' evaluation scores.
- (4) Team management: 3 points – all members on time for presentation, effective collaboration