Turgut Alp Edis
21702587 Section 2

CS315 Homework #1

Arrays in Dart
----

1. What types are legal for subscripts?

In Dart, only integer types are allowed for subscripting (indexing) the array.

2. Are subscripting expressions in element references range checked?

Subscripting expressions in arrays are range checked. Program will throw error if the index is out of the bounds of the array.

3. When are subscript ranges bound?

The arrays' lifetime is heap-dynamic in Dart. The ranges are dynamic and bound at runtime.

4. When does allocation take place?

Allocation is dynamic and in the heap. It takes place at runtime.

5. Are ragged or rectangular multidimensional arrays allowed, or both?

Dart supports both ragged and rectangular multidimensional arrays.

6. Can array objects be initialized?

The array objects can be initialized both with function and implicit way since the arrays are implicit heap-dynamic in Dart.

7. Are any kind of slices supported?

The slicing is possible with sublist method in Dart.

8. Which operators are provided?

The array operations are not provided in Dart.

Arrays in JavaScript
----

1. What types are legal for subscripts?
Only integers are legal for subscripts.

2. Are subscripting expressions in element references range checked?
In JavaScript, the index is checking whether it is in the range. However, when the index is out of range, it returns undefined instead of error because it will expand the array according to the given index number.

3. When are subscript ranges bound?
The arrays in JavaScript are heap-dynamic, the binding of subscript ranges is dynamic, and ranges are bound at runtime.

4. When does allocation take place?
The array lifetime is heap-dynamic, so allocation of the array is dynamic and take place at runtime.

5. Are ragged or rectangular multidimensional arrays allowed, or both?
JavaScript allows both ragged and multidimensional arrays.

6. Can array objects be initialized?
Array objects can be initialized when the array object is creating because the array is implicit heap dynamic.

7. Are any kind of slices supported?
Slicing is supported with slice method in JavaScript.

8. Which operators are provided?
Array operators are supported in JavaScript with map function.

Arrays in PHP

----

1. What types are legal for subscripts?
PHP arrays can have either integers or strings as subscripts.

2. Are subscripting expressions in element references range checked?
Range is checking. If the index is out of bound, it will return undefined offset error.

3. When are subscript ranges bound?
PHP supports heap-dynamic arrays therefore binding of subscript ranges is dynamic and bound at runtime.

4. When does allocation take place?
PHP supports heap-dynamic arrays therefore storage allocation is dynamic and can change any number of times and it takes place at runtime.

5. Are ragged or rectangular multidimensional arrays allowed, or both?
PHP supports rectangular arrays but not ragged arrays.

6. Can array objects be initialized?
Array objects can be initialized in PHP.

7. Are any kind of slices supported?
PHP supports slices with array_slice function.

8. Which operators are provided?
PHP arrays supports union, equality, identity, inequality and non-identity operators for arrays.

Arrays in Python

Since Python has not any built-in array data type, numpy module is inspected for python arrays.
1. What types are legal for subscripts?
Only integer, range and slice types are legal for subscripting (indexing) the array.

2. Are subscripting expressions in element references range checked?
Range checking is supported in Python. If the range is out of the index of the array, the error will be thrown.

3. When are subscript ranges bound?
The arrays in Python is heap-dynamic, then ranges are dynamic and bound at run-time.

4. When does allocation take place?
In Python, the allocation is heap-dynamic, storage allocation is dynamic and takes place at runtime.

5. Are ragged or rectangular multidimensional arrays allowed, or both?
Both ragged and rectangular multidimensional arrays are allowed in Python as array of array.
However, in the ragged array, the types will become list instead of numpy.ndarray.

6. Can array objects be initialized?
Yes, the array objects can be initialized when the array is created.

7. Are any kind of slices supported?
Python support slices as if the data in the array is accessed.

8. Which operators are provided?
In Numpy, all vector operators (+,-,/,*) are provided.

Arrays in Rust

1. What types are legal for subscripts?
Only integers is legal for subscripts.

2. Are subscripting expressions in element references range checked?
Only integers is legal for subscripts.

3. When are subscript ranges bound?
The arrays in Rust are static and stored in stack and ranges are bound at runtime.

4. When does allocation take place?
The allocation is static stack. It is bound at runtime.

5. Are ragged or rectangular multidimensional arrays allowed, or both?
Only rectangular multidimensional array is supported in Rust.

6. Can array objects be initialized?
Array objects can be initialized when the array is creating.

7. Are any kind of slices supported?
Slicing is supported.

8. Which operators are provided?
It is unknown whether the arrays support array operators or not.

Which language is the best for array operations, for various criteria?

Among 5 different programming languages, the arrays is evaluated in terms of design problems which are the diversity of subscript types, whether the range is checked or not, the binding time of subscript ranges, the time of allocation, the allowance of rectangular and ragged multidimensional arrays and the supported array operators. The diversity is up to different supported index type, and it is scored according to that number. Whether the range is checked is scored 1 if it is true, 0 otherwise. The binding time of subscript ranges is scored 1 if it is bounded in runtime and score will increase if the time is earlier than runtime. The time of allocation is scored 1 if it is bounded in runtime and score will increase if it is in earlier times. The allowance of the multidimensional arrays is scored from 0 if neither of them is supported to 2 if both of them is supported. Finally, the supported array operations is scored according to the amount of the operations. When the languages are evaluated according to these criteria, Dart is scored 6, JavaScript is scored 10, PHP is scored 11, Python is scored 12 and Rust is scored 5. Therefore, according to the criteria implemented from design problems, the best language for array operations is Python.

How is this task completed? My Learning Strategy and Approach to This Homework

When I started the homework, initially I read the first, which aims to introduce the language, and array documentation of some languages since fortunately, I did tutorship in Bilkent Python course whose code is CS115, doing the Python part and answering design questions of Python was not required to read these documentations for me. Also, I learnt the fundamentals of Dart language and wrote some example codes before, so I did not need to read the documentation of Dart. However, since I had almost no knowledge about JavaScript, PHP and Rust, I had to read the documentation of these three languages. For learning strategy, I read the documentation from the official websites for documentation such as MDN Web Docs, then I took a look at example codes from the lecture examples found on Dijkstra server and found on the educational websites such as W3Schools to learn how to code the specific language. After reading the documentations, I tried to implement the code to answer the design issues. Even if I received unexpected error or could not implement the code correctly, I searched the Stackoverflow for these issues and solved the error or wrong implementation problem. For instance, for JavaScript, I used MDN Web Docs (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array?retiredLocale=tr, https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/A_first_splash) for arrays and introduction to this language. For PHP, I used W3Schools (https://www.w3schools.com/php/php_arrays.asp, https://www.w3schools.com/php/) for arrays and introduction. For Rust, I used official documentation website of Rust (https://doc.rust-lang.org/rust-by-example/primitives/array.html) for arrays, introduction and multidimensional arrays. Since I had to learn just small part of the language, which is array data type and how to compile and run the code, the learning time is relatively short. However, before writing the code, I needed to learn how to run the compile and run the code, also what packages I need since without that knowledge, I cannot compile and run the code. After gaining this knowledge, I started to write the code for explain the design problem questions. I used Visual Studio Code to write and run the code. Besides, I used terminal to run the codes because the extensions sometimes occurred with an error. Since I used Ubuntu as Operating System, working with terminal is easier than Windows since Ubuntu allows users to work more freely. While I was writing the code, I run the code regularly to test answers of the design problem questions. I tried to run possible answers for each question, and I tried to find correct answer. Though, I cannot demonstrate answers for questions 3 and 4, which I took their answers from textbook and class notes, I could show other answers for questions. Also, I tested the corner cases of the questions as possible. For instance, for question 2, I gave the index number bigger than the number of elements in array and check if there is an error. After I finished the code, I copied it to the Dijkstra server and tested it in the server also to ensure that the code is working properly. When the code was compiled and run truly, I adjusted the comment sections to indicate the which line belongs to which question. Then, the code part of the homework finished.