

CME 3202 - Concepts of Programming Languages

Laboratory Worksheet#5

March 29, 2023

Laboratory Aim

In this laboratory section, you are expected to exercise examples of the sixth chapter in your lecture book - *"Data Types"*.

Subrange Types

"A subrange type" is a contiguous subsequence of an ordinal type. Some programming languages, such as Pascal and Ada, allow subrange types.

Exercise 1

Ada is a programming language, which allows subrange types. Apply the following steps, and implement a sample illustrating subrange types.

Step 1

Go to the website - <http://ideone.com/>

Step 2

Select Ada programming language. Click on the text editor, and add the following lines.

```
with Ada.Integer_Text_IO, Ada.Text_IO;
use Ada.Integer_Text_IO, Ada.Text_IO;

procedure Test is
  subtype Input_number is Integer range 0..10;
  Input : Input_number;
  type Colors is (Red, Green, Blue, Magenta, Cyan, Yellow);
  subtype Primary_colors is Colors range Red..Blue;
  Selected_color : Colors;
  Subcolor : Primary_colors;
begin
  Get(Input);
  Selected_color:= Red;
  if Selected_color in Subcolor then
    Put ("Primary Color ");
    New_Line;
  else
    Put ("Secondary Color");
    New_Line;
  end if;
  Put(Input);
  New_Line;
end;
```

Step 3

Write input value as 110. Click *"Run"* button, and paste and **explain** the output in the following.

```
Secondary Color
110
```

Although Input_number is defined as a subrange between 0..10, entering 110 still prints because the value is not directly validated against the range by the Ada runtime in this case. Since Input is not used in any operation except for printing, the program does not crash or throw an error, even with an out-of-range input.

Step 4

Write input value as 10 and paste the output **then change** the value of *Selected_color* to **Cyan** in the code. Click **“Run”** button, and paste the output in the following. Explain the reasons.

Secondary Color

10

The *Selected_color* was set to Cyan, which does **not** fall within the *Primary_colors* subrange (which includes Red, Green, and Blue). Hence, the condition *if Selected_color in Subcolor* evaluates to **false**, and the program prints "Secondary Color" accordingly. The input value 10 is simply printed afterward as per the code.

Step 5

What are the main advantages of subrange types?

Subrange types restrict valid values, which enhances type safety and prevents errors. They make the intent of the variable clearer to programmers. They can help detect invalid data at compile time or runtime. May result in performance optimization through smaller memory footprints. Improve code readability and documentation by clearly defining expected ranges.

Heterogeneous Arrays

“A heterogeneous array” is the one in which the elements need not be the one of the same types. Some programming languages, which support these arrays, are Perl, Python, JavaScript and Ruby.

Exercise 2

Perl allows heterogeneous arrays. Apply the following steps, and implement a sample illustrating heterogeneous arrays.

Step 1

Go to the website - <http://ideone.com/>

Step 2

Select Perl programming language. Click on the text editor, and add the following lines.

```
#!/usr/bin/perl

@string = ("One", "Two", "Three");

push(@string, "Four");
print "@string\n";

push(@string, True);
print "@string\n";

@string[2] = 4;
print "@string\n";

pop(@string);
print "@string\n";
```

Step 3

Click **“Run”** button, and paste the output in the following.

```
One Two Three Four
One Two Three Four True
One Two 4 Four True
One Two 4 Four
```

Step 4

Which property of JavaScript allows the implementation of heterogeneous arrays?

JavaScript supports heterogeneous arrays due to its dynamically typed nature and loosely typed variables. Arrays in JavaScript are actually specialized objects, allowing them to store elements of any data type (strings, numbers, objects, etc.) within the same array without restriction.

Slices

"A slice of an array" is some substructure of that array. Some programming languages which support slices are FORTRAN, Perl, Python, Ada and Ruby.

Exercise 3

FORTRAN supports slices. Apply the following steps, and implement a sample illustrating slices.

Step 1

Go to the website - <http://ideone.com/>

Step 2

Select FORTRAN programming language. Click on the text editor, and add the following lines.

```
program sliceArray
  integer, dimension(10,10) :: theArray
  integer :: i, j

  theArray = 0
  theArray(3:7,3:7) = 1

  do i = 3,7
    do j = 3,7
      theArray(i,j) = j
    end do
  end do

  do i = 1,10
    do j = 1,10
      write(*,"(I2)",advance="no") theArray(i,j)
    end do
    write(*,*)
  end do
  print *, theArray(:, 4) ! fourth column
  print *, theArray(7, :) ! seventh row
  print *, theArray(:7, 3:7)
  print *, theArray((/3,4,5/), :)
end program sliceArray
```

Step 3

Click **"Run"** button, and paste the output in the following.

Output:

```

0000000000
0000000000
0034567000
0034567000
0034567000
0034567000
0034567000
0034567000
0000000000
0000000000
0000000000
  0    0    4    4    4    4    4    0    0    0
  0    0    3    4    5    6    7    0    0    0
  0    0    3    3    3    3    3    0    0    4    4    4    4
4    0    0    5    5    5    5    5    0    0    6    6    6    6
6    0    0    7    7    7    7    7    0    0    0    0    0    0
    0    0    0    0    0    0    3    3    3    4    4    4    5
5    5    6    6    6    7    7    7    0    0    0    0    0    0
0    0    0

```

Step 4

What does `:` mean in `theArray (: 7)`?

The `:` is a range operator. `:7` means “from the **beginning** (1) to 7”.

So `theArray(:7, 3:7)` selects:

Rows **1 through 7**,

Columns **3 through 7**.

It represents a submatrix from the top part of the array.

Step 5

What is `/` used for in `theArray ((/3, 4, 5/), :)`?

`(/3,4,5/)` is an explicit array constructor in FORTRAN.

It allows selecting specific rows (3rd, 4th, 5th) rather than a continuous range.

So `theArray(/3,4,5/), :)` selects:

Rows 3, 4, and 5,

All columns.

This is useful when you want non-contiguous slices or a specific subset.