Test 2

Programming Language Concepts

April 20, 2020

Name: HUNTER DURAND

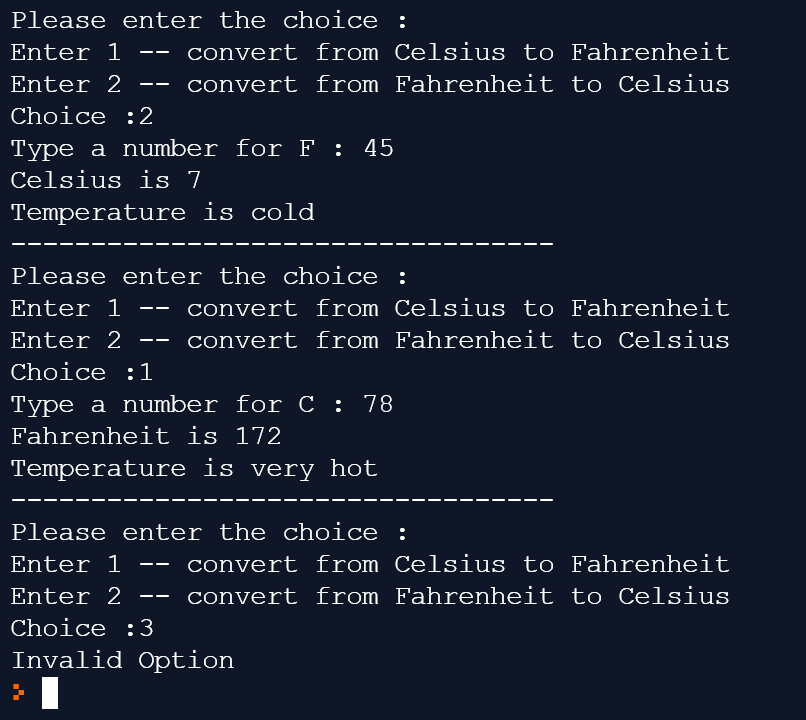
1

Programming Language Concepts Test 2

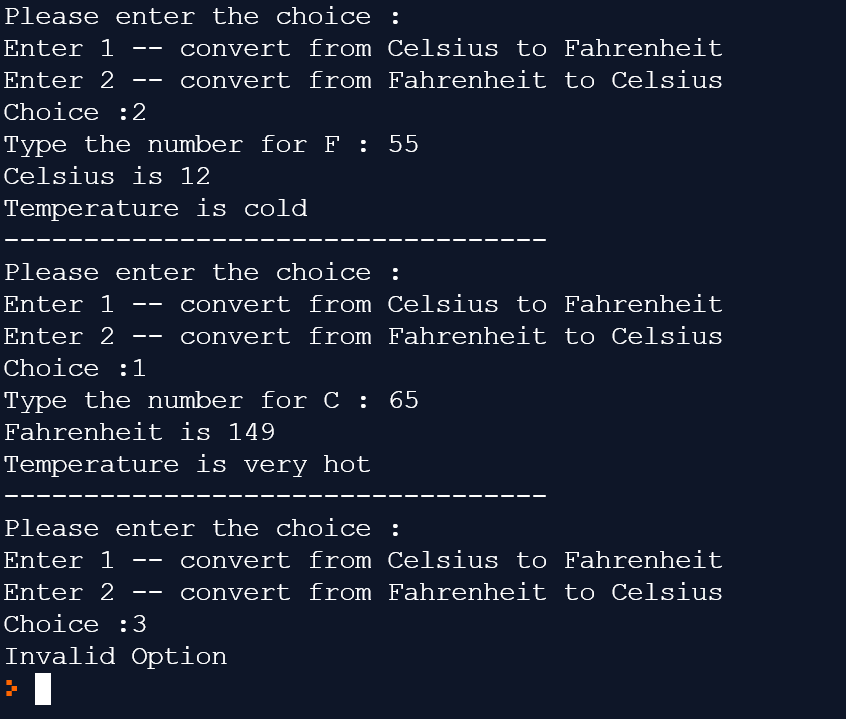
1. (15 points) Write a program in C++ or C that includes two different enumeration types and has a significant number of operations using the enumeration types. Also write the same program using only integer variables.

Compare the readability and predict the reliability differences between the two programs.

Part a. **Source Code file Name :**Question 1A.C++file

**Output**: 

Part B **only integer variable Source Code File Name:** Question 1B.C++file

**Output:** 

**Explanation:** Enum is defined as a value type data type which is used to declare a list of named integer constants and can be defined using the enum keyword inside a class, or structure. The enum is used to give a name to each constant so that the constant integer can be called using its name. In most cases enum provides a better readability than showing some integer values. So, program1 is more reliably and redability is better then program 2.

1. (10 points) Perl allows both static and a kind of dynamic scoping. Write a Perl program that uses both and clearly shows the difference in effect of the two. Explain clearly the difference between the dynamic scoping described in this chapter and that implemented in Perl.

Online Perl Compiler:[https:*//www.tutorialspoint.com/execute\_perl\_online.php*](https://www.tutorialspoint.com/execute_perl_online.php)

**Source Code File Name:** Question2

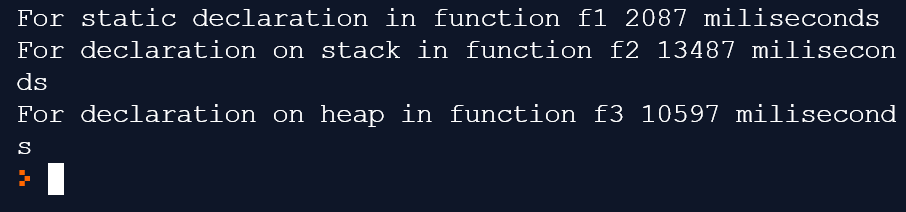
**Output:**

**Explanation:** Static scoping also known as lexical scoping. It includes If as a variable name’s scope for a certain function, then its scope is the program text of the function definition: within that text, the variable name exists, and is bound to its variable, but outside that text, the variable name does not exist. Dynamic scoping differs, it includes if as a variable name’s scope for a certain function, then its scope is the time-period during which the function is executing. While the function is running, the variable name exists, and is bound to its variable, but after the function returns, the variable name does not exist.

1. (10 points) Write three functions in C/C++: one that declares a large array statically, one that declares the same large array on the stack, and one that creates the same large array from the heap. Call each of the subprograms a large number of times (at least 100,000) and output the time required by each. Explain the results.

Explain why or why not you can’t do this in Java, what are the implication of this? If you can’t specify which type of array can you not declare.

**Source Code File Name:** Question3 arrayTimes

**Output:**

**Explanation:** From the time required to allocate and deallocate variables is often insignificant on today's machines. One can observe this from the information gathered from the program's output. Managing the heap and obtaining references to variables from the heap is costly and complicated. Therefore, creating arrays of the same size from the heap requires a significant less amount of time in comparison to declaring a static array. So based on this, in java you cannot carry out this function because all array memory allocations are from heap memory and we cannot mention an array explicitly from stack.

1. (10 points) Convert front.c and rda.c given in the content area of chapter 4; specifically the functions for EXPR, TERM, and FACTOR given in to Java; Add a working function for error and the Modolo operation in its proper order of operations.

Describe how this represents its precedence and associativity rules.

1. (10 points) Let the function fun be defined as

**int** fun(**int** \*k) {

\*k += 4; **return** 3 \* (\*k) - 1;

}

**void** main() {

**int** i = 10, j = 10, sum1, sum2;

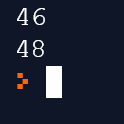
sum1 = (i / 2) + fun(&i);

sum2 = fun(&j) + (j / 2);

}

Run the code in on some system that supports C and edit it to determine the values of sum1 and sum2. Explain the results. Also, explain the results if there were no precedence rules.

**Source Code File Name:** Question5 Cfile

**Output:** 

**Explanation:** For C++by default codes are executed left >right only by default in this condition. So, in the condition provided when evaluated,

left > right,

If Xi=10 FUN(&i)return value of 41 and sets k to 14

Sum1= (10/2)+41,or 46

Sum2 = 41+ (10+2) or 48

Right > left

Sum1 = (14/2) + 41 = 48

Sum2 =41+(10/2) = 46

Meanwhile, for Java it will be the same except how the pointers are used will differ. For example, in Java it will take the pointers as a class or a referenced variable.

INT \*k in java can be used as,

Class value

{

Value k;

K =newOb();

6. (12 points) Consider the following program, written in **JAVASCRIPT-LIKE** syntax::

*// main program* var x, y, z; function sub1() { var a, y, z; . . .

} function sub2() { var a, b, z; . . .

} function sub3() { var a, x, w; . . .

}

Page 2 of 3 67 points

Programming Language Concepts Test 2

Given the following calling sequences, what reference environments of the last subprogram activated at the line that contains the ellipses ( . . . )? Include with each visible variable the name of the unit where it is declared. Provide one answer for dynamic scoping rules and one answer for static scoping rules.

1. main calls sub1; sub1 calls sub2; sub2 calls sub3.
2. main calls sub1; sub1 calls sub3.
3. main calls sub2; sub2 calls sub3; sub3 calls sub1.
4. main calls sub3; sub3 calls sub1.
5. main calls sub1; sub1 calls sub3; sub3 calls sub2.
6. main calls sub3; sub3 calls sub2; sub2 calls sub1

**Explanation:** Answer below starting with a-f; for dynamic scoping rules.

1. a x w in sub3; b, z in sub2; y in sub1

b. a x w in sub3; y z in sub1

c. a y z in sub1; x w in sub3; b in sub2

d. a y z in sub1; x w in sub3

e. a b z in sub2; x w in sub3; y in sub1

f. y z in sub1; b in sub2; x w in sub3

**Explanation:** Answer below staring with a-f; for static scoping rules.

a. sub3 refers: a x w defined in sub3; y z defined in main

b. sub3 refers: a x w defined in sub3; y z defined in main

c.sub1 refers: a y z from sub1; x from main

d. sub 1: visible variables

a y z of sub1; x of main

e. sub2: visible variables

a b z of sub2; x y of main

f. sub1 variables: a y z of sub1; x of main

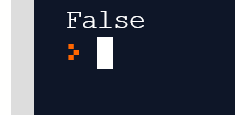
1. (8 points) Evaluate *a>b>c* in terms of mathematics (logical inequalities. Evaluate the same expression in terms of a C based language. Do the two mean the same thing? What does each expression say? If the two are the same show how they are the same or different.

7a. Source File Name: Question7A in terms of logical inequalities

Output: 

**Explanation:** Here in if condition a>b checks the values and if a is more than b then it return true and then checks b>c if b is more than c then return true so both are true then && operator return true. So if condition become true then print if block. Print the True.

7b. **Source File Name:** Question7B in terms of C based language.

**Output:** 

**Explanation:** The > operator associates left to right, so the in if condition works differently like if( (a>b) >c ). And this is the main problem. C does not have a boolean type. It uses int. So, a>b return 0 of int. Then check 0<c so it is false. One true and one false then && operator gives the output 0. So, if condition does not execute and else block will be executed. That's why false is printed.

1. (15 points) Assume the following rules of associativity and precedence for expressions:

Precedence

Highest

- (unary), (prefix) ++, (prefix) --

(postfix) ++, (postfix) --

+, \*, &

-, /, %, not

< , <=, >=, >, !=

=, +=, \*=, /=

and or,xor

Lowest

Associativity

Left to right

Show the order of evaluation of the following expressions by parenthesizing all subexpressions and placing a superscript on the right parenthesis to indicate order. For example, for the expression a \* b + c / d the order of evaluation would be represented as

((a \* b)ˆ1 + c)2 / d)3

Also rewrite the expression where there are no precedence rules and the statement is given right to left associativity, For example for the same problem above

d / c + b \* a

If you feel the statement can’t be rewritten without parenthesis explain why?

1. a \* b - 1 + c
2. ++a \* (b - 1) / c % d
3. (a - b) / c & (d \* e / a - 3)
4. -a or c = d and e
5. a > b xor c or d <= 17

**Explanation:** Answer below staring with A-E.

1. \* and + has same priority so then: ((a \* b) - (1+c)).
2. For (++c++) = ((++c)++) , prefix has high precedence.

In remaining operators \* and mod has same precedence So evaluate from left to right. ((a \* (b - 1)) / (((++c)++) mod d))

There for order of evaluation: ((a \* (b - 1)) / (((++c)++) mod d))

1. for (d \* e / a - 3), \* has high priority so ((d \*e) / a - 3). Now / and - have the same priority so then see association from left to right. so

And & has more priority than / so then order of evaluation: ((a - b) / (c & (((d \* e) / a) - 3)).

1. In question d, unary minus(-) has high priority then, - has high priority, then and has high priority and then or has low priority. So then order of evaluation: ((-a) or ((c = d) and e))
2. Here > and <= has high and same precedence then ( a > b) xor c or (d <= 17)

now xor and or has same priority so associates from left to right.

order of evaluation is: ( ((a > b) xor c) or (d <= 17))

Expression where there are no precedence rues and the statement is given right to left associativity:

a. c + 1 - b \* a

b. d mod (++c++) / (1 - b) \* b

c. (3 - a / e \* d) & c / (b - a)

d. e and d = c or -a

e. 17 <= d or c xor b > a

9. (10 points) Write a BNF description of the precedence and associativity rules defined for the expressions in problem above.

a\*(b-1)/c mod d 🡪 ((a\*(b-1)1)2/c)3mod d)4

a>b xor c or d <=17 🡪 (((a>b)1 xor c) 3 or(d<=17)2)4

As the lowest procedure is that of or and xor then

<expr> -> <expr> or <e1> | <expr> xor <e1> | <e1>

Next upper procedure includes “and” then

<e1> -> <e1> and <e2> | <e2>

Then comes =, /=, <, <=, >, >=

<e2> -> <e2> = <e3> | <e2> /= <e3> | <e3> < <e3>

|<e2> <= <e3> | <e2> > <e3> | <e2> >= <e3> | <e3>

The BNF for –(unary) will be:

<e3> -> <e4>

Next higher precedence if of, +,-,&, mod, then:

<e4> -> <e4> + <e5> | <e4> - <e5> | <e4> & <e5> | <e4> mod <e5> | <e5>

Grammar of \*, /, not will be:

<e5> -> <e5> \* <e6> | <e5> / <e6> | not <e5> | <e6>

<e6> -> a | b| c | d |e | const | (<expre>)

Page 3 of 3 33 points