SSL – Esp. Ing. Pablo Mendez
Legajo: 2035145 Apellidos: CITARELLA Nombres: TOBÍAS SANTIAGO

Actividad de clase NRO1

Instrucciones

1. Responda las siguientes preguntas, de forma manuscrita y prolija.
2. Escanee las hojas (incluído este enunciado habiendo completado sus datos en el encabezado) en un único pdf..
3. Cree un usuario en GitHub con el correo electrónico institucional FRBA asociado a la cuenta. Si ya posee una cuenta GitHub con el correo frba, puede saltar este paso.
4. Cree una carpeta K2055_SSL en su repositorio, luego cree una subcarpeta llamada Introduccion.
5. Suba el pdf y colóquelo dentro de la subcarpeta Instrucción.
6. Comparta la carpeta K2055_SSL con el profesor: buscar el usuario por su correo: pmendez@frba.utn.edu.ar.
7. Complete los datos del repositorio en la siguiente planilla:

   https://docs.google.com/spreadsheets/d/1GZF2_tbLTjzBT6EZo4SVIcjx4Fo1pVpo64huuQfDjGg/edit?usp=sharing

Preguntas contextuales de índole general

1. ¿Con qué profesor cursó Algoritmos y Estructuras de datos?
2. ¿En qué año cursó la materia?
3. ¿Tiene el final aprobado?
4. ¿Qué lenguajes de programación ha utilizado, ya sea académicamente o de manera profesional?

Preguntas relacionadas con el contenido de la materia

5. ¿Sabe qué es un identificador? Explique.
6. ¿Cómo podría especificar de manera genérica una sentencia de asignación como las vistas en AyED? (Asignación Interna).
7. ¿Sabe qué es un valor-L o L-Value? Dé tres ejemplos diferentes.
8. En AyED, ¿qué tipo de dato utilizó para el manejo de archivos?
9. ¿Conoce la diferencia entre un archivo de texto y un archivo binario? Dé una definición de no más de dos renglones de qué es un archivo de texto.
10. Dé ejemplos de expresiones vistas en AyED
11. ¿Qué tipos de sentencias (proposiciones si usa K&R en castellano) ha visto en AyED, mencione al menos 4.
12. Busque la especificación de este tipo de sentencias en el K&R e indique cómo se expresan. Ayuda: Lo puede ver en el apéndice A.

Referencia

Kernighan, B. W., & Ritchie, D. M. (1991). *El lenguaje de programación C* (2da ed.).

1) Diego Juan.

2) 2023

3) No, nos Dugo el final aprobado.

4) Académicamente he utilizado C++ y por fuera de lo académico JavaScript.

5) Sí se lo que es un identificador. Los identificadores son los nombres utilizados para referenciar las variables, las funciones, las etiquetas y otros objetos definidos por el usuario.

6) Una sentencia de asignación es una instrucción que modifica el valor de una variable. Se utiliza para almacenar valores en variables o constantes.

7) Sí, se lo que es un L-Value. Es una expresión que representa memoria. Ejemplos:

```
a = 1;          int x;          int arr[3];
└ LValue        int *p;         arr[0] = 5;
                p = &x;         └ L Value
                *P = 35;
                └ LValue
```

8) Para el manejo de archivos utilizamos File*.

9) Sí, conozco la diferencia entre un archivo binario y uno de texto. El archivo de texto se utiliza para almacenar cadena de caracteres. En cambio el binario, se utiliza para almacenar cualquier tipo de dato.

Archivo de texto: Almacena datos en formato de caracteres legibles, generalmente codificados en ASCII.

10)
```
if (x > 10) {            i = 0;
  cout << "Hola mundo";  while (i < 5) {
}                          cout << "hola mundo";
                         }
```

11) En AyED, vimos sentencias de iteración, sentencias de selección, de salto y de expresión

└ Cualquier función — While — IF — salto: BREAK

12) Sentencia de expresión: La expresión del lado derecho se evalúa y su resultado se asigna a la variable del lado derecho izquierdo, que debe ser un L-Value válido.

proposición-expresión:
expresión;

Sentencia de selección: La expresión debe valuarse a un valor entero. Si es distinto de 0, se ejecuta la sentencia IF, de lo contrario, se ejecuta la del else (Si existe)

↑ Para if

IF (expresión) proposición
IF (expresión) proposición else proposición
Switch (expresión) proposición

Switch → Debe valuarse a un tipo entero (int, Char, ...). Cada case debe tener una constante entera única. El break exita que la ejecución continue con el siguiente case. El default es opcional

Sentencia de iteración: La subproposición se ejecuta en forma repetida mientras que el valor de la expresión sea ≠ de 0, la expresión debe tener tipo aritmético o puntador. (While y do)
Con While, la prueba, incluyendo todos los efectos colaterales de la expresión, ocurre antes de c/ ejecución; con do, la prueba ocurre después de c/ iteración

Proposición-de-iteración:
While (expresión) proposición
do proposición while (expresión);

(For): La primera expresión se evalúa una vez y especifica inicialización para el ciclo. No hay restricción en cuanto a su tipo. La segunda expresión debe tener tipo aritmetico o apuntador (Si es = 0 se termina el For). La tercer expresión se evalúa antes de c/ iteración y especifica una reinicialización. No hay restricción en cuanto a su tipo.

For (expresión1 ; expresión2 ; expresión3) proposición

Sentencia compuesta : de salto : (return) : termina la ejecución de una función, opcionalmente devuelve un valor

(Break) : Sale del bucle o sentencia switch

(continue) : Salta a la siguiente iteración del bucle

Proposición-de-salto :

goto identificador ;
continue ;
break ;
return expresión ;