

Ejercicios de Programación - Sebesta

Lenguajes de Programación - ESPOL

29 de enero de 2014

1. Introducción

Las respuestas propuestas en este repositorio son producto del trabajo de los estudiantes de la materia “Lenguajes de Programación” de la ESPOL, correspondientes a las preguntas del libro de Robert Sebesta, Concepts of Programming Languages.

2. Preguntas y Respuestas

2.1. Capítulo 5: Nombres, Enlaces y Alcances.

2.1.1. **Pregunta 4: Escriba un script en Python con subprogramas triplemente anidados, donde cada subprograma referencie a variables que han sido definidas en otros niveles de la jerarquía**

Escriba su respuesta con claridad. En las secciones de código utilice listings. Capítulo 5 pregunta 5 En caso de querer llamar en nuestro programa números pseudoaleatorios, que han sido generados por un subprograma se necesita del history-sensitivity para almacenar la última iteración de la variable localmente. Ejemplo de un subprograma generando numeros pseudoaleatorios:

```
public void generarSerieDeAleatorios () {  
  
    Random numAleatorio;  
    numAleatorio = new Random ();  
    for (int i=0; i < serieAleatoria.size(); i++) {  
        serieAleatoria.set(i, numAleatorio.nextInt(1000) );  
    }  
}
```

2.2. Capítulo 7: Expresiones y Asignaciones de sentencias.

2.2.1. **Pregunta 1:** Ejecute el código dado en el Problema 13 (Set de Problemas) en un sistema que soporte C para determinar los valores de sum1 y sum2. Explique los resultados.

Código dado en el Problema 13 (Set de Problemas)

```
int fun(int *k){
    *k+=4; /* Incrementa el valor de x + 4 */
    return 3 * (*k) - 1 /* Retorna */
}

void main(){
    int i=10, j=10, sum1, sum2;
    sum1 =(i /2) + fun(&i );
    sum2 = fun(&j ) + (j /2);
}
```

OUTPUT:

sum1 = 46

sum2 = 48

Estos resultados se da por las Reglas de Asociatividad del lenguaje C, que es la asociatividad de izquierda a derecha, esto significa que el operador de la izquierda se evalúa primero.

Para sum2, la dirección de memoria cambia debido a las operaciones que se realiza con la referencia a la variable fun, por esta razón hay diferentes resultados.