

FINAL LENGUAJES DE PROGRAMACIÓN I – 2da Febrero 2017

CRITERIO DE CORRECCIÓN: PARA APROBAR EL EXAMEN EL ALUMNO DEBE COMO MÍNIMO:

- CALIFICAR CON BIEN EL EJERCICIO 1 O EL EJERCICIO 2
- NO PUEDE TENER MAL NI EL EJERCICIO 1 NI EL EJERCICIO 2
- NO PUEDE CALIFICAR CON MAL DOS EJERCICIOS EN TODO EL EXAMEN

Real int Double
Ejercicio 1. Dada la asignación $a := b + c$ donde las variables están declaradas como *real* a , *int* b y *double* c , complete el cuadro justificando todas sus respuestas.

	a) ¿Dónde se almacenan las variables?	b) ¿Se usan variables auxiliares?	c) ¿Cuáles variables cambian su tipo?	d) ¿Hay pérdida de precisión?	e) ¿Qué se hace en compilación/interpretación y en ejecución?
a) Lenguajes estáticos					
b) Lenguajes tipo Algol					
c) Lenguajes dinámicos					

Ejercicio 2. El lenguaje RUST posee el concepto de "Ownership", el cuál permite, a modo de ejemplo, que cuando se ejecuta la función F se crea un vector ' v ', el cual se almacena una parte en la pila y otra parte en el heap. Cuando finaliza la ejecución de la función F , el vector ' v ' desaparece, tanto del heap como de la pila.

```
fn F() {
    let v = vec![1, 2, 3];
}
```

- a) ¿Qué tipo de variable es v ? Proporcionar un ejemplo de otro lenguaje que posea la misma característica. Considerando que dicho lenguaje tiene la característica de poder evitar que se asigne la dirección de ' v ' a otra variable, responder:
- b) ¿Es posible que el tiempo de vida de ' v ' exceda el de la función $F()$?
- c) ¿Es posible que el tiempo de vida de ' v ' sea menor que el de la función $F()$?

Fundamentar todas las respuestas.

Ejercicio 3. En el siguiente programa escrito en el lenguaje Java:

```
class Base {
    public int x=1;
    void m() {System.out.println(x);}
}
class Derivada extends Base {
    public int x=2;
    void m() {System.out.println(x);}
}
public class Final {
    public static void main(String []args){
        Base b = new Base(); //1
        System.out.println(b.x); //2 → 1
        Derivada d = new Derivada(); //3
        b = d; //4
        System.out.println(b.x); //5 → 2
        b = new Derivada(); //6
        System.out.println(b.x); //7 → 2
        d = b; //8
        b.m(); //9 → 2
        d.m(); //10 → 2
    }
}
```

- a) Indicar, para cada una de las líneas del código indicadas si se produce un error de compilación, un error de ejecución, o ninguno de los dos
- b) En el caso que no se produzcan errores, describir la semántica de las asignaciones que contiene el programa y para las sentencias de impresión, qué salida producen.
- c) En //9 y //10 ¿se ejecuta el mismo código?

Si se produjesen errores de compilación, ignorar la línea y continuar con el análisis. Fundamentar las respuestas.

Ejercicio 4. Se tiene una función en lenguaje C, donde los arreglos se pasan por referencia, y cuyo parámetro es un arreglo donde sus elementos están almacenados por filas:

```
static int w=8;
//9 int x[1][3] {
    x[1][3]=1;
}
//10 p() {
    int y[3][w];
    g(0);
    w=7;
    q();
}
//11 q() {
    int y[3][w];
    g(0);
}
//12 f() {
```

Responder los siguientes ítems justificando sus respuestas:

- a) ¿Es posible resolver en compilación la asignación al elemento del arreglo? Si su respuesta es afirmativa indique cómo hace el lenguaje para encontrar el `x[1][3]`; si es negativa, indique por qué lo es y cómo se soluciona.
- b) ¿Qué parte de la asignación al elemento del arreglo se resuelve en compilación y qué parte en ejecución?
- c) ¿Se produce algún error de ejecución o comportamiento inesperado durante la ejecución?
- d) Suponiendo que ahora el lenguaje implementa el pasaje de parámetros por copia valor ¿Qué diferencias existen con el pasaje de parámetros por referencia, considerando el inciso a)?