

## FINAL LENGUAJES DE PROGRAMACIÓN I – Septiembre 2019

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

- CALIFICAR CON BIEN EL EJERCICIO 3 O EL EJERCICIO 4
- NO PUEDE TENER MAL NI EL EJERCICIO 3 NI EL EJERCICIO 4
- NO PUEDE CALIFICAR CON MAL DOS EJERCICIOS EN TODO EL EXAMEN

**Ejercicio 1.** Dada la asignación  $a := b + c$ ; Responda las siguientes preguntas. Justifique detalladamente todas sus respuestas

- a) Si manda el tipo de  $a$  ¿se convierte el tipo de  $b$  o se cambia el tipo de  $b$ ?
- b) Si manda el tipo de  $b$  ¿se cambia el tipo de  $a$  o se convierte el tipo de  $a$ ?
- c) ¿En qué tipos de lenguajes se dan cada uno de estos casos?
- d) ¿En qué casos se usan variables auxiliares y dónde se almacenan las mismas? ¿Estas son accedidas explícitamente por el programador o por el lenguaje?
- e) ¿En qué casos la conversión o cambio puede generar fragmentación?

**Ejercicio 2.** Considere que una función  $f$  crea dos hilos de ejecución a partir de una función  $h(x)$ , la cual lee y escribe el parámetro  $x$  declarado en  $f$ .

- a) Realizar un diagrama de memoria indicando la ubicación de las variables.
- b) Indicar para los siguientes tipos de parámetros reales y semánticas de pasaje, cuáles podrían producir problemas de resultados inesperados. En el caso de producirlos, indicar cuál es el motivo de los eventuales problemas de estos resultados, fundamentando la respuesta.
  - i. Objeto en el heap pasado por copia en C++/Pascal
  - ii. Objeto en la pila pasado por referencia en C++/Pascal
  - iii. Objeto en el heap pasado por referencia en C++/Pascal
  - iv. Puntero a objeto pasado por copia en C++/Pascal
  - v. Puntero a objeto pasado por referencia en C++/Pascal
  - vi. Puntero a variable semiestática pasado por copia en C++/Pascal
  - vii. Variable estática pasada por referencia en C++/Pascal
  - viii. Objeto estático en Java
  - ix. Objeto sin atributos estáticos en Java
  - x. Variable semiestática pasada por copia en C++/Pascal

**Ejercicio 3.** Dada la siguiente regla gramatical del lenguaje Kotlin (los no terminales se escriben en mayúscula, los elementos seguidos de un "?" pueden aparecer 0 o 1 veces, los terminales se encuentran entre comillas simples), responder si las sentencias de cada inciso son sintácticamente correctas o no, fundamentando la respuesta.

EXPRESIONIF  $\rightarrow$  'if' '(' EXPRESION ')' (CUERPO | ';' ) | 'if' '(' EXPRESION ')' CUERPO? '?' 'else' (CUERPO | ';')

- a) `if(1==0); else println("else")`
- b) `if(1==0) else println("else")`
- c) `if(1==0) println("else")`
- d) `if(1==0) println("if") else`
- e) `if (1==1) println("if")`
- f) `if (1==1)`
- g) `if (1==1);`
- h) `if(1==0) println("if") else println("else")`
- i) `if(1==0) println("if"); else println("else")`
- j) `if(1==0) println("if"); else if(1==0) println("else_if"); else println("else_else")`

Ejercicio 4. Para el siguiente programa, con pasaje de parámetros por referencia mediante alias:

```
static int x; Alias
void f(referencia int p, copiar valor int q) {
    static int y;
}
void h() {
    static __thread int w;
    int s;
    ...
    h();
    ...
}
void g() {
    int z;
    f(z,x);
    pthread_create(.....h,...);
    ...
    pthread_destroy(.....h,...);
}
void main() {
    g();
}
```

- Realizar un diagrama de memoria de la ejecución del programa indicando la ubicación de las variables
- Para cada variable especificar cuándo comienza y cuando finaliza su tiempo de vida
- ¿Cuál tipo de variable introducida, por el hecho de poseer threads, modifica el tiempo de vida de las variables existentes en un programa que no posee threads? ¿Por qué?