

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
LENGUAJES DE PROGRAMACIÓN
SEGUNDA EVALUACIÓN II TÉRMINO 2014-2015

Nombre: _____

Fecha: _____

Indicaciones:

- Leer con detenimiento las preguntas planteadas.
- Poner en práctica sus principios éticos.
- El tiempo para el desarrollo de la evaluación es de 2 horas.

1) ¿Qué imprime por pantalla el siguiente código en LISP?. (5 pts)

<pre>(defclass point () (x y z)) (setf my-point (make-instance 'point)) (type-of my-point) (defun set-point-values (point x y z) (setf (slot-value point 'x) x (slot-value point 'y) y (slot-value point 'z) z)) (set-point-values my-point (- 4 1) (* 2 2) (+ 10 2)) (defun distance-from-origin (point) (with-slots (x y z) point (sqrt (+ (* x x) (* y y) (* z z))))) (distance-from-origin my-point)</pre>	<p>a) T b) NIL c) 19 d) 13 e) Ninguna de las anteriores</p>
--	---

2) Escriba el resultado de las siguientes declaraciones. (5 pts)

a)

<pre>(setq x '(1 2)) (setq y '((3 4))) (append x (car y) (list 5))</pre>	<p>=></p>
---	--------------

b)

<pre>(setq x '(a b c)) (append x x)</pre>	<p>=></p>
---	--------------

c)

<pre>(setq x '(a b c)) (cons (caddr x) (cons (cadr x) (cons (car x) nil)))</pre>	<p>=></p>
--	--------------

3) ¿Qué imprime por pantalla el siguiente código en JAVA?. (5 pts)

<pre> class X{ int xpto(){return 2;} } class Y extends X{ int xpto(){return 7;} void test(){ X x = (X) this; System.out.print(this.xpto()); System.out.print(x.xpto()); System.out.print(((X)this).xpto()); System.out.print(super.xpto()); } public static void main(String[] args){ new Y().test(); } } </pre>	<p>a) 2 2 2 7 b) 7 2 2 7 c) 7 7 2 2 d) 7 7 7 2 e) Ninguna de las anteriores</p>
--	---

4) Complete el diagrama de jerarquía entre clases. (5 pts)

<pre> (defclass animal () ((legs :reader leg-count :initarg :legs) (comes-from :reader comes-from :initarg :comes- from))) (defclass mammal (animal) ((diet :initform 'antelopes :initarg :diet))) (defclass bear (mammal) ((cute-p :accessor cute-p :initform nil))) (defclass figurine () ((potter :accessor made-by :initarg :made-by) (comes-from :initarg :made-in))) (defclass figurine-bear (bear figurine) ((name :reader bear-name :initarg :bear-name) (diet :initform nil))) </pre>	<pre> graph BT t[t] --> standard-object[standard-object] subgraph Unknown direction TB A[] B[] end A --> standard-object B --> standard-object </pre>
---	--

5) Seleccione las respuestas correctas. (5 pts)

- a) El chequeo de tipos asegura que los operandos (parámetros) de un operador (subprograma) sean de tipos compatibles.
- b) El tipado fuerte requiere que los tipos de todos los operandos puedan ser determinados, tanto en tiempo de compilación como de ejecución.
- c) La equivalencia de tipos por nombre es más difícil de implementar pero brinda mayor flexibilidad al programador.
- d) Una de las métricas más importantes para comparar algoritmos de recolección de basura (Garbage Collection) incluye la maximización del tiempo requerido para recuperar memoria.
- e) En lenguajes de programación se denomina coerción a la conversión automática de tipos compatibles.

6) Considere estas dos posibilidades a la hora de definir las clases A y B . (10 pts)

<pre>final class A{ public void metodo1 () {...} final public void metodo2 () {} } class B extends A{</pre>	<pre>class A{ public void metodo1 () {...} final public void metodo2 () {} } class B extends A{ public void metodo2 () {} }</pre>
--	---

¿Cuál de las siguientes afirmaciones se cumple para las clases anteriores?. Justifique su respuesta.

- a) Solo la primera es incorrecta.
- b) Las dos versiones son incorrectas.
- c) Solo la segunda es incorrecta.
- d) Las dos son correctas.

7) Considere el siguiente código: (15 pts)

<pre>class Hilo extends Thread { private int id; public Hilo(int i){ id = i; } public void run(){ int i, valor = id; for(i=0; i<3; i++){ valor = valor * valor; try{ if (id <2) Thread.sleep (10); else Thread.sleep(1); }catch(InterruptedException e){} } System.out.println(valor); } }</pre>	<pre>class TestHilo{ public static void main(String[] args){ Hilo h1 = new Hilo(1); Hilo h2 =new Hilo(2); h1.run(); h2.start(); try{ h1.join(); }catch(InterruptedException e){} System.out.println("Fin"); } }</pre>
---	--

¿Cuál de las siguientes opciones muestra la salida por pantalla al ejecutar el programa?:

a) 1 Fin 256	b) 256 1 Fin	c) Fin 256 1	d) 256 Fin 1
---------------------------	---------------------------	---------------------------	---------------------------

Justifique su respuesta:

8) Indique la afirmación correcta. (5 pts)

- a) Mediante el uso de semáforos el programador no tiene que responsabilizarse de la gestión del acceso exclusivo a un recurso.
- b) Tanto con monitores como con semáforos el programador no tiene que responsabilizarse de la gestión del acceso exclusivo a un recurso.
- c) Tanto con monitores como con semáforos el programador tiene que responsabilizarse de la gestión del acceso exclusivo a un recurso.
- d) Mediante el uso de monitores el programador no tiene que responsabilizarse de la gestión del acceso exclusivo a un recurso.

9) Responda con oraciones de sentido completo. (15 pts)

- a) ¿Cuándo decimos que una función tiene efectos colaterales (side effects)?

- b) Describa los tres modelos semánticos de paso de parámetros en subprogramas.

- c) ¿Cuáles son los tipos de abstracción en lenguajes de programación?

10) ¿Para qué es utilizada la palabra reservada “friend” en C++, en funciones y clases?. Escriba un ejemplo usando clases amigas. (15 pts)

11) Escriba un programa en C que produzca comportamientos diferentes dependiendo del uso de paso de parámetros por valor o por referencia. (15pts)