

# Examen

**Objetivo:** Evaluar los conceptos de los paradigmas de programación funcional, lógico, imperativo y orientado a objetos así como su puesta en práctica.

Ingeniería en Computadores  
Lenguajes, compiladores e interpretes  
Tiempo Estimado: 120 minutos  
Profesor: Marco Rivera Meneses.

**Instrucciones:** Utilice únicamente lapicero azul o negro. El uso de lápiz o corrector es permitido pero implica la pérdida del derecho a reclamo. Cada pregunta tiene una respuesta correcta, exacta o completa. Es prohibido el uso de teléfono, tabletas o cualquier otro dispositivo electrónico. Debe desarrollar este examen en un cuaderno de examen oficial del TEC. *Cualquier intento de fraude equivale a nota cero. El examen es para desarrollo individual.*

Conteste las siguientes preguntas con el nivel mínimo y suficiente de detalle para demostrar su conocimiento del tema. No se evaluarán respuestas parciales o imprecisas.

49

Nombre: Alexis Gavriel Nota: 82

## Primera Parte: Selección Única.

1. a	2. b	3. c	4. d	5. <del>e</del>	6. <del>e</del>	7. c	8. <del>e</del>	9. b	10. <del>e</del>
11. <del>e</del>	12. <del>e</del>	13. b	14. a	15. c	16. c	17. c	18. <del>e</del>	19. <del>e</del>	20. a

1. ¿Cuál es el operador utilizado para obtener el valor almacenado en una variable puntero?

- ~~a. \*~~
- b. &
- c. &&
- d. ||
- e. Ninguno de las anteriores

2. ¿Cuál sería una expresión de punteros equivalente para referencias el elemento de un arreglo a[i] [j] [k] [l]?

- a. (((a + i) + j) + k) + l
- ~~b. \*((\*(a + i) + j) + k) + l~~
- c. (((a + i) + j) + k) + l
- d. ((a + i) + j) + k + l
- e. \*((\*(a + i) + j) + k) + l

3. ¿Cuál sería la combinación de las dos sentencias siguientes en una sola?

```
char *p;
p = (char*) malloc(100);
```

- a. char p = \*malloc(100);

- b. char \*p = (char) malloc(100);
- ~~c. char \*p = (char\*) malloc(100);~~
- d. char \*p = (char \*) (malloc\*)(100);
- e. Ninguno de las anteriores

4. Si una variable es un puntero a una estructura, ¿cuál de los siguientes operadores es usado para acceder a los miembros de la estructura?

- a. .
- b. &
- c. \*
- ~~d. ->~~
- e. &&

5. ¿Cuál es la similitud entre una estructura (structure) y una unión (union)?

- a. Ambas permiten definir nuevos valores
- b. Ambos permiten definir nuevos tipos de datos
- c. Ambos permiten definir nuevos punteros
- d. Ambos permiten definir nuevas estructuras.
- ~~e. Ninguna de las anteriores~~

6. ¿Cuáles son operadores unarios en C?

- 1. |
- 2. sizeof

3. ~  
4. &&

- a. 1,2  
b. 1,3  
c. 2,4  
d. 1,2,3

☒ Ninguna de las anteriores. ✓

7. ¿Cuál sería la salida del siguiente programa?

```
#include<stdio.h>
int main()
{
    static int a[20];
    int i = 0;
    a[i] = i ;
    printf("%d, %d, %d\n", a[0], a[1],
i);
    return 0;
}
```

- a. 1,0,1  
b. 1,1,1  
☒ c. 0,0,0  
d. 0,1,0  
e. 0,0,1

8. ¿Cuál es la salida del siguiente programa?

```
#include<stdio.h>
int main()
{
    int x=4, y, z;
    y = --x; 1
    z = x--; 1
    printf("%d, %d, %d\n", x, y, z);
    return 0;
}
```

- ☒ a. 4,3,3  
b. 4,3,2  
c. 3,3,2  
d. 2,3,3  
b. Ninguna de las anteriores

9. ¿Cuál encabezado debería incluirse para usar funciones como malloc() y calloc()?

- a. memory.h  
☒ b. stdlib.h ✓  
c. malloc.h y calloc.h  
d. stdio.h  
e. Ninguna de las anteriores

10. ¿Cuál es la salida del siguiente programa?

```
#include<stdio.h>
int main()
{
```

```
int i=0;
for(; i<=5; i++);
    printf("%d", i); 0 1 2 3 4
return 0;
}
```

- a. 0,1,2,3,4,5,  
b. 5  
c. 1,2,3,4  
d. 6

☒ Ninguna de las anteriores

11. ¿Cuál es la salida del siguiente programa?

```
public class Foo
{
    public static void main(String[]
args)
    {
        try
        {
            return;
        }
        finally
        {
            System.out.println(
"Finally" );
        }
    }
}
```

- a. Finally ☒  
☒ b. La compilación falla  
c. El código corre sin ninguna salida. ←  
d. Se produce un error en tiempo de ejecución.  
e. Ninguna de las anteriores

12. ¿Cuál sería la el código necesario en el Punto X para hacer que el código compile

```
public class ExceptionTest
{
    class TestException extends
Exception {}
    public void runTest() throws
TestException {}
    public void test() /* Punto X */
    {
        runTest();
    }
}
```

- a. No es necesario ningún código.  
b. Throws exception  
☒ c. Catch (Exception e)  
d. Throws exception (e)  
e. Throws RuntimeException

13. ¿Cuál es el valor de 'd' después de ejecutar la siguiente línea de código?

```
double d = Math.round ( 2.5 +
Math.random() );
```

- a. 2
- ☒ b. 3
- c. 4
- d. 2,5
- e. Ninguna de las anteriores

14. ¿Cuál de las siguientes llamadas a Math.max son correctas.

- 1. Math.max(1,4)
- 2. Math.max(2.3,5)
- 3. Math.max (1,3,5,7) ☒
- 4. Math.max(-1.5,-2.8f)

- ☒ a. 1,2 y 4
- b. 2,3 y 4
- c. 1,2 y 3
- d. 3 y 4
- e. Ninguna de las anteriores

15. ¿Cómo debería ejecutar el siguiente programa para que la salida sea: Arg is 2

```
public class Myfile
{
    public static void main (String[]
args)
    {
        String a = args[1];
        String b = args[2];
        String c = args[3];
        System.out.println("Arg is " +
c);
    }
}
```

- a. java MyFile 222
- b. java MyFile 1 2 2 3 4
- ☒ c. java MyFile 1 3 2 2
- d. java MyFile 0 1 2 3

16. ¿Se desea que las subclases en cualquier paquete tengan acceso a los miembros de una superclase. ¿Cuál es el acceso más restrictivo que logra este objetivo?

- a. public
- b. private
- ☒ c. protected
- d. transient
- e. Ninguna de las anteriores

17. ¿Cuál es el constructor default para la siguiente clase?

```
public class Test { }
```

- a. Test()
- b. Test(void)
- ☒ c. public Test()
- d. public Test(void)
- e. Ninguna de las anteriores

18. ¿Cuál de las siguientes líneas de código genera un error de compilación?

- a. int[] scores = {3, 5, 7};
- b. int [][] scores = {2,7,6}, {9,3,45};
- c. String cats[] = {"Fluffy", "Spot", "Zeus"};
- d. boolean results[] = new boolean[] {true, false, true};
- ☒ e. Integer results[] = {new Integer(3), new Integer(5), new Integer(8)};

19. ¿Cuáles de las siguientes son firmas validas de una interface?

- 1. private int getArea();
- 2. public float getVol(float x);
- 3. public void main(String [] args);
- a. 1 y 2
- b. 2 y 3
- c. 1 y 3
- d. Todas las anteriores
- ☒ e. Ninguna de las anteriores

20. ¿Cuál es una declaración valida en una interface?

- ☒ a. public static short stop = 23;
- b. protected short stop = 23;
- c. transient short stop = 23;
- d. final void madness(short stop);
- e. Ninguna de las anteriores

**Segunda Parte: Respuesta Breve.** Conteste las siguientes preguntas con el nivel mínimo y suficiente de detalle para demostrar su conocimiento del tema. No se evaluarán respuestas parciales o imprecisas. (10 Pts)

1. ¿Qué es una función? (1pt).

Una operación que se ha de realizar a un dato.  
↳ matemáticas o lógicas

2. ¿Qué es la inferencia de tipos? (1pt).

Cuando el tipo de dato no es declarado explícitamente por el usuario.

El compilador deduce ~~el tipo de dato~~ el tipo de variable a partir de las operaciones <sup>que se hacen</sup>.

3. Mencione los tres componentes de los lenguajes funcionales. (3pts).

a) ~~no hay variables~~  
operaciones primitivas ✓

b)

funciones ??? + / -

c)

Listas ~~de~~ <sup>objetos</sup> primitivos (átomos)

4. ¿Qué es un hecho? (1pt).

una afirmación verdadera. ~~Así como~~ se toma como una verdad...

5. ¿Cuáles son las dos formas de representar un OR lógico en Prolog? (1pt).

$\neg A1 ; A2$  ~~o~~  $A1 ; A2$

6. ¿Describa que es una variable en Prolog? (1pt).

una incógnita, un valor que no se ha encontrado una vez que se ~~ha~~ ha asignado, no varra.

7. Explique que es el alcance y la duración default de las variables en C. (2pts).

Duración: Hasta salir de la función (luego se borra la variable), puede ser ~~fuera~~ fuera de las funciones para ser global y durar hasta que se cierre el programa.  
Alcance: Hasta donde puede ser accedida la variable. no puede ser llamada desde otra función, a menos que sea global.

**Tercera Parte: Desarrollo.** Conteste las siguientes preguntas con el nivel mínimo y suficiente de detalle para demostrar su conocimiento del tema. No se evaluarán respuestas parciales o imprecisas. (30 Pts)

a) Construya una función (DrRacket) que se llame **transpuesta**, que recibe una matriz y devuelve su transpuesta. (10 pts).

Ejemplo

(transpuesta '((1 2 3)

(1 2 3)

(1 2 3))

'((1 1 1)

(2 2 2)

(3 3 3))

b) Escriba los predicados Prolog para que concatene dos listas y las devuelva en una tercer lista. Use el Cut. (10 pts).

Ejemplo

concatenar([a,b,c],[d,e],

[a,b,c,d,e]).

True

c) Plantee la estructura de clases en Java para implementar el patrón Observador. (10 pts)

a)

```

(define (transpuesta mat)
  (cond ((null? mat) mat)
        (else (append (get_col mat) (transpuesta (get-rest mat)))))
)

(define (get_col mat)
  (cond ((null? mat) mat)
        (else (append (car mat) (get_col (cdr mat)))))
)

(define (get-rest mat)
  (cond ((null? mat) mat)
        (else (append (cdr (car mat)) (get-rest (cdr mat)))))
)

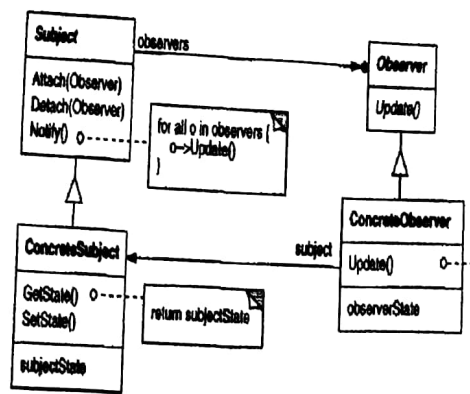
```

(8)

b)

Concatenar:  $([ ], L, L), !$  (10)

concatenar  $([A|L1], L2, [A|L3])$ : - concatenar  $(L1, L2, L3)$ .



Public <sup>abstract</sup> class Subject {

Protected Observer[] observers = new Observer[];

Protected void Attach(Observer New) {  
observers.append(New); }

Protected void Detach(Observer Ob) {  
observers.delete(Ob); }

Protected void Notify() {  
for (o in observers)  
o.Update();

10

}  
Public <sup>abstract</sup> class Observer {

Public abstract Update();

}

```
Public class Concrete Observer extends Observer {  
    Private Subject subj ;  
    Private bool observerState;  
    Public void Update () ;  
}
```

```
Public class ConcreteSubject extends Subject {  
    Private bool subjectState ;  
    Public bool GetState () {  
        return subjectState;  
    }  
    Public void SetState ( bool s ) {  
        subjectState = s ;  
    }  
}
```