

# Paradigmas de la Programación

## Recuperatorio del Primer Parcial

23 de Junio de 2022

Apellido y Nombre: \_\_\_\_\_

1	
2	
3	

4	
5	
6	

1. [10 pt.] El siguiente código tiene una instrucción **break**. Esta instrucción produce un salto a otra parte del código. Entonces, el código que usa **break**, ¿es una característica del código spaghetti? ¿por qué?

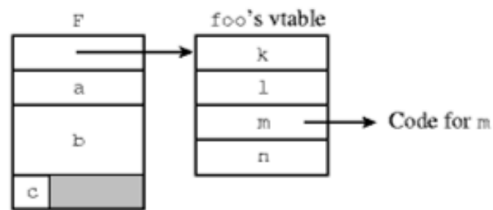
```
1  while (true) {
2      decrease(k);
3      if (k!=0) {
4          set(p,u_k);
5          increasev2(a[k]);
6          if (q != 0) {
7              break;
8          }
9      } else {
10         return;
11     }
12 }
```

2. Escriba en pseudocódigo un programa en el que:
- a) [10 pt.] una componente tenga efectos secundarios pero sea determinística
  - b) [10 pt.] una componente no tenga efectos secundarios pero no sea determinística
3. [10 pt.] En esta imagen se observa cómo C++ guarda los métodos virtuales en una vtable. Explique cómo es el proceso mediante el cual se usa la vtable en C++, explique la diferencia entre métodos virtuales y no virtuales en términos de *overhead* y *flexibilidad*, y qué decisión de diseño de C++ justifica que por defecto todos los métodos sean no virtuales.

```

class foo {
    int a;
    double b;
    char c;
public:
    virtual void k ( ...
    virtual int l ( ...
    virtual void m ();
    virtual double n( ...
    ...
} F;

```



4. [10 pt.] ¿Qué imprime el siguiente programa con alcance estático, y qué imprime con alcance dinámico? Justifique su respuesta usando los conceptos como el de *lugar de definición de la función* y el de *variable global*.

```

x : integer    -- global

procedure set_x(n : integer)
    x := n

procedure print_x
    write_integer(x)

procedure first
    set_x(1)
    print_x

procedure second
    x : integer
    set_x(2)
    print_x

set_x(0)
first()
print_x
second()
print_x

```

5. [20 pt.] Cuáles son los valores de "x" y "z" al final del siguiente bloque, asumiendo que el pasaje de parámetros es a) por valor, b) por referencia y c) por valor-resultado?

```

{ int y;
  int z;
  y := 7;
  { int f(int x) {
      x := x+1;
      y := x;
      x := x+1;
      return y
    };
    int g(int x) {
      y := f(x)+1;
      x := f(y)+3;
      return x
    };
    z := g(y)
  }
}

```

6. [30 pt.] Explique verbalmente qué va sucediendo en la ejecución del siguiente programa, ayudándose de diagramas de la pila de ejecución cuando lo considere necesario. No es necesario representar las variables locales, control links, access links, retorno de función ni ninguna otra información que no sea relevante al manejo de excepciones. Explique también qué imprime el programa.

```

1  class Program {
2      static void Main(string[] args) {
3          try {
4              try {
5                  throw new Exception("exception_thrown_from_try_block");
6              }
7              catch (Exception ex) {
8                  Console.WriteLine("Inner_catch_block_handling_{0}.", ex.Message);
9              }
10             finally {
11                 Console.WriteLine("Inner_finally_block");
12             }
13         }
14         catch (Exception ex) {
15             Console.WriteLine("Outer_catch_block_handling_{0}.", ex.Message);
16         }
17         finally {
18             Console.WriteLine("Outer_finally_block");
19         }
20     }
21 }

```