Pregunta 1: Definir 3 paradigmas de programación y dar sus características.

Paradigma imperativo

- Los programas imperativos contienen instrucciones que dicen al ordenador cómo realizar una tarea. Los primeros lenguajes imperativos fueron los códigos máquina de los ordenadores, que utilizaban instrucciones sencillas y permitían implementar el hardware fácilmente, pero no servían para desarrollar programas complejos.

- El primer lenguaje imperativo que posibilitó la creación de programas con un nivel de complejidad elevado fue FORTRAN. Hoy en día está representado por los lenguajes de programación BASIC, C ó PASCAL, entre otros.

Paradigma funcional

- Los programas funcionales se basan en el uso de una o más funciones dentro de las cuales se pueden utilizar funciones creadas anteriormente. Su objetivo es dividir el programa en módulos de forma que cada uno de éstos realice una única función.
- El primer lenguaje de programación funcional fue LISP. Existen dos tipos de lenguajes funcionales: los puros (como HASKELL) y los híbridos (SAP, ML, Scheme).

Paradigma orientado a objetos (POO)

- La programación orientada a objetos expresa un programa como un conjunto de objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener, reutilizar y volver a utilizar. Su uso se popularizó a principios de los 90 y actualmente son muchos los lenguajes de programación asociados a este paradigma.
- Las características del paradigma orientado a objetos son: encapsulamiento, abstracción, polimorfismo y herencia.
- Muchos lenguajes utilizados en la actualidad están orientados a objetos, como Java, C++, Python o Delphi. Un lenguaje completamente orientado a objetos es Smalltalk.

Pregunta 2: Cual es la diferencia entre i++ e ++i dar un ejemplo en código.

i++: primero asigno, luego incremento
++i: primero incremento, luego asigno el valor incrementado

```
i=5; // i vale 5
j=5 // j vale 5
a = i++ // a vale 5, i vale 6
b = ++j // b vale 6, j vale 6
```

Una nota importante es que si no haces asignación, es lo mismo i++ que ++i, por ejemplo.

```
i=5;
j=5;

i++; // i vale 6
++j; // j vale 6
```

Pregunta 3: Completar el cuadro con la información de PRIORIDAD, siendo 1 más prioritario que 5.

OPERADORES	PRIORIDAD	OPERADORES	PRIORIDAD
/	3	(EXPR)	1
--VAR	2	+ EXPR	2
*	3	+	3
%	3	-	3
VAR--	1	&&	4

Pregunta 4: Responder las siguientes preguntas y dar ejemplos.

a) ¿Qué significa casting en programación?

Conversión entre tipos primitivos (casting) El casting es un procedimiento para transformar una variable primitiva de un tipo a otro. También se utiliza para transformar un objeto de una clase a otra clase siempre y cuando haya una relación de herencia entre ambas.

Ejemplo: int a=1;

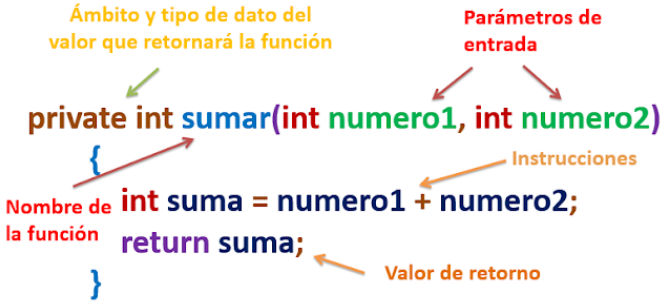
```
double b=2.5;

a=(int)b;
```

b) ¿Qué es una función y procedimiento?

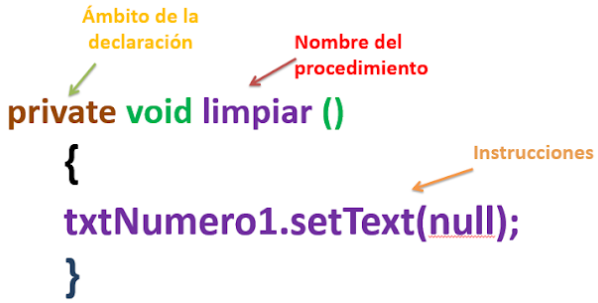
Funciones

Las funciones son un conjunto de líneas de código (instrucciones), encapsulados en un bloque, usualmente reciben parámetros, cuyos valores utilizan para efectuar operaciones y adicionalmente retornan un valor con la ejecución return.



Procedimientos

Los procedimientos son básicamente un conjunto de instrucciones que se ejecutan sin retornar ningún valor.



c) ¿Qué quiere decir sobrecarga de operadores?

Se refiere a la posibilidad de tener dos o más funciones con el mismo nombre pero funcionalidad diferente. Es decir, dos o más funciones con el mismo nombre realizan acciones diferentes.
El compilador usará una u otra dependiendo de los parámetros usados. A esto se llama también sobrecarga de funciones.

Ejemplo: `Color (int r, int g, int b)`
`Color (float a, float b, float c)`