

Respuestas a Preguntas

Lección 1

- Pregunta 1

Pregunta:

Cual es la forma correcta para utilizar **cout**

```
std::cout << "Hola Mundo" << std::endl;
```

Los signos de menor están correctamente dispuestos.

94% contestó correctamente

- Pregunta 2

Pregunta:

Vincular los tipos del mundo de especificación y los del lenguaje C++.

1. Números naturales: **int**
2. Números reales: **float**
3. Booleanos: **bool**
4. Caracteres: **char**

100% contestó correctamente

- Pregunta 3

```
int main() {  
    // declaracion de las variables  
    float x = 3.0;  
    int y = 8, z;  
  
    // operacion  
  
    z = y / x; // z recibe el valor resultado de la operación, que al convertirlo a entero da 2  
    // salida por consola  
    std::cout << z; // muestra 2 por consola
```

```
return 0;  
}
```

100% contestó correctamente

- Pregunta 4

Supongamos tenemos estas líneas de código:

```
float x = 3.0;  
int y = 8, z;  
bool test;  
test = (x = 5) && (y < 10); // (x=5) NO es una fórmula lógica porque tiene un solo '='. Es una  
asignación. TODAS las asignaciones deberían ser exitosas, entonces (x=5) es VERDADERO.  
Luego (y<10) es VERDADERO, con lo que test es VERDADERO.
```

Luego de correrlas en un programa, test=falso?

La respuesta es FALSO.

69% respondieron correctamente.

- Pregunta 5

Sea una variable:

```
float n;
```

Cuál sería la construcción correcta para aprovechar la lógica de cortocircuito de los operadores AND y OR en C++?

Rta: `(n != 0 && 1.0 / n)` // si n=0 no se evalúa el 2do miembro de la fórmula y no se define el 1/n

99% contestó correctamente.

- Pregunta 6

Se quiere definir una función que se llame "calculoPromedio", de tipo flotante, que devuelva el promedio entre tres números flotantes que se reciben como parámetros de entrada.

Elegir la declaración correcta:

Rta: `float calculoPromedio(float valor1, float valor2, float valor3);` // los parámetros van separados por comas y delante de cada uno debe estar indicado el tipo de variable que es.

El 86% contestaron correctamente

- Pregunta 7

Declarar una función de tipo booleana que verifique que verifique si una entrada de caracter por consola es una 's'. La función se tiene que denominar "sigoPreguntando".

Rta: `bool sigoPreguntando(char valor);` // el nombre de la función es la pedida y el tipo de variable recibido por parámetro. El nombre no importa. El tipo de la variable se escribe delante del nombre de la variable, como cuando la definimos dentro de un cuerpo de código. No se pueden usar para los nombres de variables palabras reservadas, como "if", "else", "int", "char", etc.

94% contestó correctamente

- Pregunta 8

Siempre tiene que existir un cuerpo de código para cuando la guarda es falsa.

Es FALSO. El "else" puede obviarse si no es preciso tener un código cuando la guarda de falso.

El 88% contestó correctamente

- Pregunta 9

```
int main()
{
    int i = 0;
    for(i=2;i<3;i=i+1)
        std::cout << i << std::endl; // este ciclo se ejecuta solo una vez, con el valor de i=2, ya que
    // luego de imprimir por consola a i, el for aumenta en 1 el valor de i, como dice el tercer
    // término. Ahora i=3 y esto hace falsa la guarda, con lo cual no ejecuta el cuerpo de código y
    // sale del for.
    return 0;
}
```

92% eligieron la respuesta correcta

- Pregunta 10

- ```
int main()
{
 int i = 0;
 int valor = 0;
 // imprimo los valores entre floatntes entre 1.0 y 10.0
 while(i < 10) {
 std::cout << valor << std::endl;
 valor++; // incremento en 1 valor
 }

 return 0;
}
```
- Encuentran un problema?

El problema era que la guarda nunca se hacía falsa, ya que no se variaba el valor de i dentro del cuerpo de código del while. Esto hace que entre en un bucle infinito, y el programa se cuelga.

Aquellas cosas que digan los comentarios, solo son comentarios. No son precisamente instrucciones que deban seguirse al pie de la letra.

84% contestaron correctamente

## Lección 2

- Pregunta 1

```
void sumarMasUno(int i) {
 int k = 1;
 i = i + k;
}

int main()
{
 int k = 5;
 int i = 8;

 cout << "Llamo a la funcion" << endl;
 sumaMasUno(k); // k = 5, i = 8. La función no modifica los valores de las variables de main
 cout << k << endl; // muestra 5
 cout << i << endl; // muestra 8
 return 0;
}
```

88% contestaron correctamente

- Pregunta 2

```
int sumarMasUno(int i) {
 int k = 1;
 i = i + k;
 return i;
}

int main()
{
 int k = 5;
 int i = 8;

 cout << "Llamo a la funcion" << endl;
 i = sumaMasUno(k); // k es igual a 5, pero i toma el valor de lo que devuelve sumarMasUno
 cout << k << endl; // muestra 5
 cout << i << endl; // muestra 6
 return 0;
}
```

66% contestaron correctamente

- Pregunta 3

Pregunta:

Para el caso de precisar enviar a una función un elemento de datos extenso (por ejemplo una fotografía color de 12 mega pixeles), desde el punto de vista de tiempo de ejecución y espacio de memoria, me conviene utilizar:

**En el video habla sobre esto, e indica que la mejor opción es por REFERENCIA.**

**96% contestaron correctamente**

- Pregunta 4

Pregunta:

No existe un problema de seguridad en el paso por referencia, ya que si la función a la que se llamó modifica los datos de la función que hizo la llamada, estos pueden ser recuperados ya que quedan en memoria.

Si existe un problema de seguridad en el paso por referencia. Si los datos que son pasados a una función son modificados, ya no los puedo recuperar desde el programa principal (a menos que haya tenido el cuidado de duplicarlos). La respuesta es FALSO.

89% contestaron correctamente

- Pregunta 5

Pregunta:

Una función devuelve UNA sola variable.

Si yo quisiera calcular dentro de una función el área y el perímetro de un triángulo rectángulo, a partir de la base y altura del mismo, la definición correcta que me permitiría hacer el cálculo es:

- a) float calcularAreaYperimetro(float base, float altura, float & area);
- b) void calcularAreaYperimetro(float base, float altura, float & area, float & perimetro);

AMBAS son correctas. La opción (a) devuelve un flotante que podría ser el perímetro, y recibe como parámetro una variable por referencia donde puede almacenar el cálculo del área para que sea recuperado por el programa principal. La opción (b) hace esto con dos variables por referencia.

26% contestó correctamente

- Pregunta 6

```
void prueba(int x, int &y) {
 x = x + y;
 y = x - y;
 x = x - y;
}
```

Así como está planteado, la respuesta es **a**, al utilizar un llamado **prueba(a,a)**, sin embargo, esta pregunta tiene un error en el enunciado. El encabezado de la misma debería ser

void prueba(int **&**x, int &y)

El sistema esperaba el 0, que era al final erróneo. No tomamos en cuenta esta pregunta.