

## Parcial C++

Nombre: \_\_\_\_\_ Grupo: \_\_\_\_\_ Fecha: \_\_\_\_/\_\_\_\_/\_\_\_\_

Duración: 3 horas Total: 100 pts equivalente a 5

Temas: Nivel 1 — Variables/Tipos/IO, Operadores, Condicionales, Bucles; Nivel 2 — Funciones

### Instrucciones

- Usa C++17 o superior.
- Lee cuidadosamente cada pregunta. Marca una sola opción en la Sección A.
- En la Sección B, escribe exactamente lo que se imprime (respeta espacios y saltos de línea).
- En la Sección C, implementa solo las funciones solicitadas. No uses librerías fuera de `<iostream>` y `<string>`.

### Sección A — Opción múltiple (10 × 3 pts = 30 pts)

1) ¿Cuál declara y define una variable entera con valor 10?

- ☐ A) `int x; x == 10;`
- ☐ B) `int x = 10;`
- ☐ C) `int x( ); x = '10';`
- ☐ D) `int x { "10" };`

2) ¿Qué imprime? `std::cout << 7/2 << " " << 7/2.0;`

- ☐ A) 3 3
- ☐ B) 3 3.5
- ☐ C) 3.5 3.5
- ☐ D) 3 3.0

3) Con `bool a=true, b=false;` ¿Qué imprime? `std::cout << (a&&b) << " " << (a||b) << " " << (!b);`

- ☐ A) true true false
- ☐ B) 0 1 1
- ☐ C) 1 1 0
- ☐ D) false true true

4) Dado `int x=3; if (x=0) { ... } else { ... }` ¿qué ocurre?

- ☐ A) Error de compilación por asignación en `if`.
- ☐ B) Entra al `if` siempre.
- ☐ C) Entra al `else` porque el valor es 0 tras la asignación.
- ☐ D) Comportamiento indefinido.

5) `do-while` respecto a `while`:

- ☐ A) Nunca ejecuta si la condición es falsa.
- ☐ B) Ejecuta al menos una vez.
- ☐ C) Ejecuta exactamente una vez.
- ☐ D) Es más rápido por definición.

6) ¿Cuál lee dos enteros separados por espacio?

- ☐ A) `cin << a << b;`
- ☐ B) `cin >> a, b;`
- ☐ C) `cin >> a >> b;`
- ☐ D) `cin (a, b);`

7) ¿Qué imprime?

```
int a=5, b=2;
```

```
std::cout << a + b*3 << '\n';
```

- ☐ A) 11
- ☐ B) 21
- ☐ C) 10
- ☐ D) 7

8) ¿Cuál firma pasa un entero por referencia?

- ☐ A) void f(int n&);
- ☐ B) void f(int& n);
- ☐ C) void f(ref int n);
- ☐ D) int& f(int n);

9) ¿Qué hace continue en un for?

- ☐ A) Termina por completo el bucle.
- ☐ B) Salta a la siguiente iteración.
- ☐ C) Reinicia el contador a 0.
- ☐ D) Salta dos iteraciones.

10) ¿Cuál condición es verdadera?

- ☐ A) (3 == 3) && !(2 > 5)
- ☐ B) (3 = 3) || (2 < 1)
- ☐ C) !(3 == 3) || (2 > 5)
- ☐ D) (3 != 3) && (2 <= 5)

## Sección B — Predice la salida (5 × 6 pts = 30 pts)

Escribe exactamente lo que se imprime (respeta espacios y saltos de línea).

1)

```
int a=5, b=2;
```

```
std::cout << a + b*3 << std::endl;
```

Respuesta: \_\_\_\_\_

2)

```
int n=0;
```

```
while (n<3) {
```

```
    std::cout << n << " ";
```

```
    n++;
```

```
}
```

Respuesta: \_\_\_\_\_

3)

```
int x=10;
```

```
if (x%2==0) std::cout << "par";
```

```
else      std::cout << "impar";
```

4)

```
int i=3;
do {
    std::cout << i--;
} while (i>0);
```

5)

```
int a=1;
int b = (a++ > 1) ? 100 : 200;
std::cout << a << " " << b;
```

### Sección C — Implementación de funciones (4 × 10 pts = 40 pts)

Programa solo las funciones pedidas. Puedes probar mentalmente con casos de ejemplo.

C1) Firma: `int max3(int a, int b, int c);`

Devuelve el mayor de tres enteros usando if/else.

---

C2) Firma: `int sumaRango(int a, int b);`

Devuelve la suma de todos los enteros en [a, b] (se asume  $a \leq b$ ) usando un bucle.

---

C3) Firma: `bool contieneDigito(int n, int d);`

Devuelve true si n (no negativo) contiene el dígito d (0–9). Usa operadores aritméticos.

---

C4) Firma: `void tablaMultiplicar(int n);`

Imprime la tabla de n del 1 al 10 con el formato exacto: `n x i = resultado` (un renglón por i).