1. ¿Qué es un programa?

Un programa es un conjunto de instrucciones escritas en un lenguaje de programación que una computadora puede ejecutar para realizar una tarea específica. Estas instrucciones pueden incluir operaciones aritméticas, manipulaciones de datos, y control del flujo de ejecución.

2. ¿Cuál es la estructura básica de un programa en C++?

La estructura básica de un programa en C++ incluye: Directivas de Preprocesador: Generalmente, la inclusión de bibliotecas estándar.

#include <iostream>

Función Principal: El punto de entrada del programa.

int main() {

// Código del programa

return 0;

}

3. ¿Cuál es la diferencia entre una instrucción y una declaración en C++?

Declaración: Introduce una variable, función, o tipo al compilador sin asignar necesariamente una acción. Por ejemplo, declara una variable o una función.

int x; // Declaración de una variable

Instrucción: Realiza una acción, como asignar un valor, realizar una operación o llamar a una función.

x = 5; // Instrucción de asignación

4. ¿Cuáles son los tipos de datos básicos en C++?

Tipo de Dato Descripción Ejemplo de Uso

int Entero int a = 10;

float Número de punto flotante float b = 3.14;

double Número de punto flotante doble double c = 2.7182818284;

char Carácter char d = 'A';

bool Booleano (verdadero o falso) bool e = true;

void Sin tipo de dato, usado en funciones void func() {}

5. ¿Qué son las variables en C++ y para qué se utilizan?

Las variables en C++ son espacios de memoria con un nombre asociado que se utilizan para almacenar datos que pueden cambiar durante la ejecución del programa. Se utilizan para guardar y manipular valores.

6. ¿Qué es el alcance de una variable en C++?

El alcance (o "scope") de una variable en C++ define la región del código donde la variable es accesible. Puede ser:

Local: Dentro de una función o bloque específico.

Global: En todo el programa desde el punto de declaración hasta el final del archivo.

7. ¿Qué son las variables locales y las variables globales en C++?

Variables Locales: Declaradas dentro de una función o bloque y solo accesibles dentro de ese ámbito.

void func() {

int x = 10; // Local

}

Variables Globales: Declaradas fuera de cualquier función y accesibles desde cualquier parte del programa.

int y = 20; // Global

void func() {

y = 30; // Accede a la variable global

}

8. Operadores básicos en C++

Operador Descripción Ejemplo de Aplicación

+ Suma int a = 5 + 3;

- Resta int b = 5 - 3;

\* Multiplicación int c = 5 \* 3;

/ División int d = 5 / 3;

% Módulo int e = 5 % 3;

= Asignación int f = 5;

== Igualdad bool g = (5 == 3);

!= Desigualdad bool h = (5 != 3);

> Mayor que bool i = (5 > 3);

< Menor que bool j = (5 < 3);

9. ¿Qué son las estructuras condicionales en C++ y para qué se utilizan?

Las estructuras condicionales en C++ son constructos que permiten tomar decisiones en el flujo de ejecución del programa basándose en condiciones evaluadas en tiempo de ejecución. Se utilizan para ejecutar diferentes bloques de código dependiendo de ciertas condiciones.

Tipos de estructuras condicionales:

if: Ejecuta un bloque de código si la condición es verdadera.

if (condición) {

// Código a ejecutar si la condición es verdadera

}

if-else: Ejecuta un bloque de código si la condición es verdadera, de lo contrario, ejecuta otro bloque.

if (condición) {

// Código a ejecutar si la condición es verdadera

} else {

// Código a ejecutar si la condición es falsa

}

else if: Permite probar múltiples condiciones.

if (condición1) {

// Código a ejecutar si condición1 es verdadera

} else if (condición2) {

// Código a ejecutar si condición2 es verdadera

} else {

// Código a ejecutar si ninguna condición es verdadera

}

switch: Selecciona una de las muchas posibles ramas de ejecución basándose en el valor de una variable.

switch (variable) {

case valor1:

// Código a ejecutar si variable == valor1

break;

case valor2:

// Código a ejecutar si variable == valor2

break;

default:

// Código a ejecutar si ninguna de las anteriores condiciones es verdadera

}

10. ¿Qué son las estructuras repetitivas en C++ y para qué se utilizan?

Las estructuras repetitivas en C++ (también conocidas como bucles) se utilizan para ejecutar repetidamente un bloque de código mientras se cumple una condición específica, o para iterar sobre una serie de elementos.

Tipos de estructuras repetitivas:

for: Itera un bloque de código un número específico de veces.

for (inicialización; condición; incremento) {

// Código a ejecutar en cada iteración

}

Ejemplo:

for (int i = 0; i < 10; ++i) {

std::cout << i << std::endl;

}

while: Ejecuta un bloque de código mientras la condición especificada es verdadera.

while (condición) {

// Código a ejecutar mientras la condición sea verdadera

}

Ejemplo:

int i = 0;

while (i < 10) {

std::cout << i << std::endl;

++i;

}

do-while: Similar a while, pero garantiza que el bloque de código se ejecute al menos una vez, ya que la condición se evalúa al final del bucle.

do {

// Código a ejecutar al menos una vez y mientras la condición sea verdadera

} while (condición);

Ejemplo:

int i = 0;

do {

std::cout << i << std::endl;

++i;

} while (i < 10);