**+Instituto Superior de Formación Técnica Nº 151 logo151-trans.gif  
Carrera: Analista de Sistemas  
1 Año. Algoritmos y Estructuras de Datos I**

|  |  |
| --- | --- |
| **Trabajo Práctico Nº 2.3** | **Unidad 2** |
| **Modalidad:** Semi-Presencial | **Estratégica Didáctica:** Trabajo Grupal. |
| **Metodología de Desarrollo:** acordar | **Metodología de Corrección:** acordar docente |
| **Carácter de Trabajo:** Obligatorio – con Nota | **Fecha Entrega:** A confirmar por el Docente. |

**Unidad 2 – Instrucciones Tipos Operadors**

**Marco Teórico:**

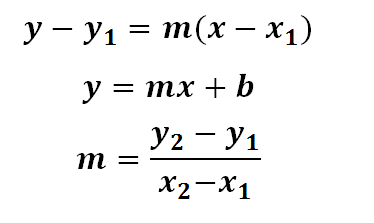
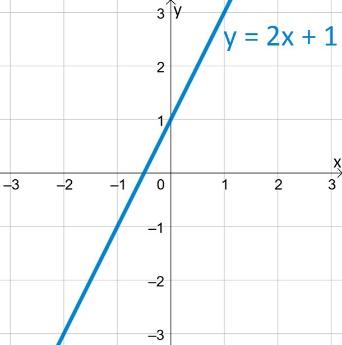
1. Qué entiende x tipo de Dato y qué establece.
2. Describir los distintos Tipos de Datos, dar ejemplos.
3. ¿Qué significa que el Lenguaje es “Case Sensitive”?
4. ¿Qué tipos de comillas llevan los Datos de tipo String?
5. ¿Qué es un Modificador de Tipo? ¿Para qué sirve?
6. ¿Dónde se ubican los Identificadores Descriptivos en las declaraciones de variables?
7. ¿Qué significa que el “Compilador” reserva Memoria Suficiente para las Variables?
8. Las variables no se inicializan Automáticamente en C++, ¿qué significa?
9. Dar ejemplos de Asignación de Variables de Distintos tipos
10. Dadas las Var A y B, ¿por qué necesito una variable Auxiliar para Intercambiar los Valores?
11. ¿Qué diferencia tiene un Operador “Monario2 de uno “Binario”?
12. Dar Ejemplos de División Real y entrera.
13. Dar ejemplos de Operadores de Incremento y Decremento.
14. Ampliar la respuesta con Prefijo y Postfijo
15. Describir el Orden de Evaluación en las Operaciones
16. Conceptualizar las abreviaturas Aritméticas
17. Explicar el Desbordamiento
18. ¿Qué es una constante? Explicar su uso.

**Marco Práctico:**  
Desarrollar en C++ los siguientes programas:

1. Resolver y Realizar en c++   
   1. Realice un programa que lea de la entrada estándar los siguientes datos de una persona: Edad: dato de tipo entero. Sexo: dato de tipo carácter. Altura en metros: dato de tipo real. Tras leer los datos, el programa debe mostrarlos en la salida estándar.
   2. La Nota final de un estudiante es el promedio de tres notas: la nota de laboratorio que cuenta un 30% del total, la nota teórica que cuenta un 60% y la nota de Práctica que cuenta el 10% restante. Escriba un programa que lea de la entrada estándar las tres notas de un alumno y escriba en la salida estándar su nota final.

|  |  |  |
| --- | --- | --- |
| **Forma** | **Fórmula** | **Variables** |
| Cuadrado | *P* = 4 *s* | *s* es la longitud del lado del cuadrado. |
| Rectángulo | *P* = 2 *L* + 2 *W* | *L* y *W* son las longitudes de los lados del rectángulo (longitud y ancho). |
| Triángulo | *a* + *b* + *c* | *a* , *b* , y *c* son las longitudes de los lados. |

* 1. Realizar un programa que pida los datos necesarios y Calcule los Volúmenes:

* 1. Nos piden un programa que resuelva la siguiente ecuación con salida x consola:  
     
  2. Nos piden un programa que resuelva la siguiente función lineal con salida x consola, deber pedir 4 valores para x dando le valor de y por consola.:  
     

Lic. Oemig José Luis.

1. Tipo de datos: Conjunto específico de valores de datos, junto con un conjunto de operaciones en esos valores.
2. Los distintos tipos de datos soportados por C++ son:

Algunos tipos de datos se emplean con tanta frecuencia que C++ los define para nosotros.

Ejemplos de estos tipos estándar (o integrados) son int (para trabajar con números enteros), float (para trabajar con números reales que tienen puntos decimales) y char (para trabajar con datos de caracteres).

Adicionalmente, C++ permite a los programadores definir sus propios datos: tipos definidos por el programador (o definidos por el usuario).

1. Que el lenguaje sea “Case Sensitive” significa que C++ diferencia entre mayúsculas y minúsculas. Es decir que las siguientes variables son distintas para el compilador de C++:

int numero, Numero, nUmero, NUMERO;

1. Los tipos de dato string se escriben entre comillas dobles.
2. Los tipos básicos pueden tener **modificadores** opcionales, que se usan para alterar el significado del tipo base (en cuanto a rango de valores y espacio de almacenamiento), de forma que dan lugar a variantes que se adecuan a ciertas necesidades específicas. Estos modificadores indican: con signo, sin signo, largo y corto, y se aplican con las palabras clave que se señalan: largo (long), corto (short), con signo (signed), sin signo (unsigned).

**Ejemplo**:

int i1; short int i2; long int i3; signed int i4;

unsigned int i5; signed short int i6;

unsigned short int i7;

signed long int i8;

unsigned long int i9;

|  |  |  |
| --- | --- | --- |
| **Especificadores de enteros** (sinónimos en la misma línea) | | |
| **int** | **signed int** |  |
| **unsigned** | **unsigned int** |  |
| **short** | **short int** | **signed short int** |
| **unsigned short** | **unsigned short int** |  |
| **long** | **long int** | **signed long int** |
| **unsigned long** | **unsigned log int** |  |

|  |  |  |
| --- | --- | --- |
| **Tipo** | **Tamaño bits** | **Rango** |
| **int** | 32 | -2,147,483,648 <= X <= 2,147,483,647 |
| **signed short** | 16 | -32768 <= X <= 32767 |
| **unsigned short** | 16 | 0 <= X <= 65535 |

|  |  |
| --- | --- |
| **Especificadores de carácter** (sinónimos en la misma línea) | |
| **char** | **signed char** (si **char** es signed por defecto). |
| **unsigned char** |  |
| **char** | **unsigned char** (si **char** es unsigned por defecto). |
| **signed char** |  |

1. Sintaxis:

[signed|unsigned] int <identificador> ;

1. Dado que C++ es un lenguaje compilado, toda variable y estructura debe declararse antes de la ejecución. Es así que el compilador reserva memoria para todas las variables declaradas antes de ejecutar.
2. Las variables no tienen un valor pre-asignado en C++ sino que debeasignarles un valor el programador.
3. Ejemplos de Asignación de Variables:

int a = 10;

int b;

b=12;

int c;

cout << “Ingrese un valor numérico: “ << endl;

cin >> c ;

1. Dadas las Var A y B, ¿por qué necesito una variable Auxiliar para Intercambiar los Valores?

Porque en caso contrario perderíamos uno de los dos valores.

Es como tener dos vasos: uno de agua y otro de aceite y querer intercambiar sus contenidos: si no contamos con un tercer vaso para trasvasar los líquidos, no tendremos posibilidad de llevar a cabo el intercambio.

1. Un operador **monario** actúa sobre un único operando. Por ejemplo, el operador lógico de negación ( ! ) el cual se aplica a **una** proposición (afirmación). Los operadores **binarios** actúan o vinculan **dos** operandos. Por ejemplo, el operador aritmético suma ( + ) vincula dos valores numéricos.
2. Dar ejemplos de división real y entera.

División real:

10.0 / 6 // resultado: 1.666666…...

10 / 6.0 // resultado: 1.666666…...

10.0 / 6.0 // resultado: 1.666666…...

División entera:

10 / 6 // resultado: 1

1. y 14. Operadores incremento ( ++ ) y decremento ( -- ):

Las instrucciones de expresión:

r++; // Post-Incremento

s--; // Post-Decremento

Son equivalentes a:

r = r + 1;

s = s - 1;

Y también se pueden escribir como:

++r; // Pre-Incremento

--s; // Pre-Decremento

15. Para describir cómo funciona el pre y post incremento, damos un ejemplo:

int i, x;

i = 2;

x = ++i;

// Ahora i = 3, x = 3

i = 2;

x = i++;

// Ahora i = 3, x = 2

16. Existen en C++ diferentes formas de abreviar una asignación:

Asignación Abreviatura

a = a + 12; a += 12;

a = a \* 3; a \*= 3;

a = a - 5; a -= 5;

a = a / 37; a /= 37;

a = a % b; a %= b;

17. Explicar el Desbordamiento

Como ya se ha visto, de acuerdo al tipo de variable que se declara, el comilador reserva un determinado tamaño de memoria. Además, también debido al tipo y los modificadores del tipo, existe para esa variable un determinado rango. Por ejemplo, vimos que una variable de tipo

short int funciona correctamente en el rango -32768 a 32767

Luego, si hacemos:

short int i = 32767; // Valor máximo para short int

i++; // 32768 no cabe en un short int

cout << i; // Muestra -32768

18. Una constante no es más que una variable inicializada a la que no le permitimos variar.

Modelo de Declaración:

const [declaracion de variable inicializada];

**Ejemplo:**

const short int Meses = 12;