**Instituto Superior de Formación Técnica Nº 151   
Carrera: Analista en Sistemas  
1 Año. Algoritmos y Estructuras de Datos.**

| **Trabajo Práctico Nº0** | **Unidad 0** |
| --- | --- |
| **Modalidad:** Semi -Presencial | **Estratégica Didáctica:** Trabajo individual. |
| **Metodología de Desarrollo:** Det. docente | **Metodología de Corrección:** Via Classroom. |
| **Carácter de Trabajo:** Obligatorio – Con Nota | **Fecha Entrega:** A confirmar por el Docente. |

**Marco Teórico:**  
Responder el siguiente cuestionario en función de la bibliografía Obligatoria.  
Bienvenidos a este Tramo formativo.  
Usaremos la Plataforma como medio de Conexión "Oficial" para poder avanzar desde lo Académico mientras dure la emergencia, luego será un vínculo más para poder estar más interconectados, sirviendo de "Soporte" a las clases presenciales.  
Comenzaremos con tareas de Lectura, Comprensión y resolución de Consignas sobre los temas propuestos a modo de Inducción en la materia.  
  
Por Favor Leer el Material y Responder las Siguientes consignas.  
  
1. Definir el concepto de programa de computadora.  
2. Que entiende por algoritmo.  
3. Que es un lenguaje de programación de alto nivel.  
4. Describir los procesos de compilación y ejecución.  
5. Brevemente Describir la historia del lenguaje C++.  
6. Para los recién iniciados, describir los principales componentes de una computadora y cómo funcionan juntos.  
7. Por qué debemos aprender acerca de algunos asuntos éticos básicos que enfrentan los profesionales de la computación.  
8. Podrías Listar las etapas básicas relacionadas con la escritura de un programa de computadora.  
9. Describir lo que es un compilador y lo que hace.  
10. Para los recién iniciados, diferenciar entre harreare y software.  
11. Describir un método apropiado de resolución de problemas para desarrollar una solución algorítmica a un problema.  
  
Utilizar el Libro "Programación-y-resolución-de-problemas-con-c++" Cap. 1  
Bibliografia - Bjarne Stroustrup - El lenguaje de programación C++  
Que será el material "Obligado" de este año.  
  
.

La Cátedra.   
Lic. Oemig José Luis.

## Answers

* 1. Un programa es una *secuencia lógica de instrucciones escritas* (algoritmos) *en un lenguaje de programación*. El objetivo de tal secuencia es realizar una tarea, ya sea controlar el hardware, hacer cálculos o procesar datos.
* 2. Un algoritmo es una *secuencia de pasos definidos, ordenados y finitos* que se siguen para resolver un problema. Es independiente del medio en el que se emplea y el formato. Un algoritmo puede ser la receta para hacer un café, los pasos para cambiar el aceite del auto, o las instrucciones de un robot para moverse.
* 3. Un lenguaje de alto nivel es un lenguaje de programación diseñado para ser facil de leer y escribir por humanos. La sintaxis es similar al lenguaje natural, #Cpp, #CSharp y #C son algunos ejemplos. Estos lenguajes abstraen los detalles de la comunicación con el #hardware , precisando un intérprete o compilador que los traduce a código máquina.
* 4.
  + Compilación: El compilador traduce el código fuente de alto nivel a código objeto o código máquina. En el proceso incluye, la revisión de sintaxis, expansión de macros y el anexo (o fusión) de las direcciones de las bibliotecas referenciadas por el código fuente, usando el enlazador. El producto final es un archivo binario o ejecutable como punto de entrada para la ejecución del programa.
  + Ejecución: Con el entorno de ejecución adecuado, el archivo ejecutable se carga en memoria y la CPU ejecuta instrucciones secuencialmente, interactuando con el OS y con el hardware, dependiendo de cada instrucción.
* 5.
  + 1980: Bjarne Stroustrup desarrolla C++ en los laboratorios Bell, extendiendo el lenguaje C con características de programación orientada a objetos #OOP (como clases y herencia).
  + 1985: Primera versión comercial.
  + 1998: Se estandariza (*C++98*).
  + Actualmente sigue evolucionando (*C++11*, *C++17*, *C++20*), añadiendo funcionalidades modernas.
* 6.
  + CPU: Ejecuta instrucciones y procesa datos.
  + Memoria RAM: Almacena temporalmente datos y programas en uso.
  + Disco Duro/SSD: Almacenamiento permanente.
  + Tarjeta Madre: Conecta todos los componentes.
  + Periféricos: Entrada/Salida (teclado, monitor, etc.)
  + FUNCIONAMIENTO: La CPU accede a datos de la Memoria RAM o Disco Duro/SSD, los procesa y envía resultados a Periféricos, coordinado por el sistema operativo.
* 7. El conocimiento de la ética en la industria del software se ha vuelto fundamental para evitar las siguiente problemáticas en individuos y/o sociedades:
  + Dilemas con la privacidad de los datos
  + Problemas de seguridad cibernética / digital.
  + Sesgo en los algoritmos entrenados.
  + Distintos impactos del software en la sociedad.
* 8. Las etapas básicas para escribir un programa consisten en:
  + *Fase de Resolución*:
    - Análisis: Entender qué debe resolver el programa.
    - Diseño: Planear la lógica (pseudocódigo/diagramas).
    - Verificación: Comprobar si el algoritmo resuelve el problema.
  + *Fase de implementación*:
    - Desarrollo: Escribir el código en un lenguaje.
    - Prueba: Verificar que el programa funcione correctamente.
  + *Fase de Mantenimiento*:
    - Ejecución: Utilización del programa en contexto real.
    - Mantenimiento: Corregir errores o mejorar.
  + Documentación: Es transversal a todas las fases de producción de un programa.
* 9. Un compilador es un programa que traduce código fuente a código objeto o código máquina (ejecutable), verificando errores de sintaxis, optimizando el código y/o usando el enlazador del entorno de desarrollo para anexar o fusionar las *librerías referenciadas* en las instrucciones. El archivo generado es un binario o ejecutable.
* 10. Los aspectos que pueden diferenciar el software y hardware, pueden resumirse en:
  + Hardware: Referido a componentes físicos (CPU, Disco Duro/SSD, teclado, placa madre, etc)
  + Software: Todo lo referido a programas e instrucciones que controlan el hardware (sistema operativo, aplicaciones)
* 11. Un método para la resolución de problemas mediante algoritmos puede definirse en los siguiente pasos:
  + Análisis y Especificación: Entender (*definir*) el problema a resolver y especificar lo que *DEBE* hacer la solución (que se espera que realice el *algoritmo*). Implica también definir las *entradas* del sistema y las *salidas* esperadas.
  + Solución General (Algoritmo): Desarrollo de la secuencia de pasos que resuelve dicho problema. En esta etapa se pueden usar distintos métodos, como dividir el problema en subproblemas (*dividir y vencer*), *resolución por analogía*, etc.
  + Verificación: *Seguir* los pasos del *algoritmo*, de forma *ordenada y precisa*. Asegurarse de que el sistema *realiza correctamente la tarea esperada*. Se comprueba por cada entrada que las salidas sean las que se especificaron en la primera etapa.