



**UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES**

Programación II

Realizado por:

Christian Antonio Viramontes Juárez

Flujos de entrada y salida de datos (cin/cout)

Flujos de entrada y salida de datos(cin/cout)

1. Programa que imprima "hola mundo" utilizando cout.
2. Programa que imprima el valor de un string dentro de cout.
3. Programa que imprima dos string con cout que son separados por un salto de línea.
4. Programa que reciba un valor string con cin y que imprima lo recibido con un cout.
5. Programa que recibe dos números y muestra el resultado de los números sumados.
6. Programa que reciba un numero integer y lo imprima. Se tendrá que validar que si es un numero en caso de no serlo tendrá que pedirlo hasta que si sea un numero valido.
7. Programa que captura y imprima:
 - Nombre completo
 - Dirección
 - Experiencia profesional.

Se tendrá que utilizar cin.getline para la captura de las 3 variables y con la ayuda de cin.getline permite que en la experiencia profesional se termine de

escribir con # en vez de con un enter del teclado. Esto va a permitir que se podrá dar salto de líneas al momento de capturar la experiencia profesional.

Manejo de memoria dinámica

Operadores new y delete

8. Programa que recibe dos números de entrada y muestre la dirección de las variables en la memoria junto con el valor de las variables.
9. Programa que declare un puntero donde se utiliza el operador new para aloclarle memoria de un tipo int y que imprima la dirección.
10. Programa que reciba un numero donde se guarda un integer en un pointer y que lo imprima. El pointer se declara y se guarda memoria con el operador new. Al final se debe utilizar el operador delete para borra el puntero de la memoria.
11. Programa que declara un array de tipo integer de tamaño 10 con el operador new. Se captura un valor para cada espacio del array. Después se imprimirá el valor de cada espacio del puntero junto con la dirección que ocupa en la memoria. Finalmente se borra el array de la memoria con el operador delete.

Arreglos Dinámicos

12. Programa que captura un entero y se declare un array dinámico del tamaño introducido. Se captura un valor entero para cada posición del array. Finalmente imprima el array completo y bórralo de la memoria.

13. Declara un array dinámico donde el usuario ingresa el tamaño del array. Se captura un valor para cada tamaño y se tendrá que comparar todos los números del array para finalmente imprimir el mas alto y su posición en el array. Finalmente se tendrá que borrar el array de la memoria.
14. Programa que lea un numero y declara un array dinámico del tamaño leído. Se llenará aleatoriamente con números entre 1 y 10. Finalmente imprime el array resultante con su media.
15. Programa que pida las dimensiones de una matriz y declara una matriz dinámica de las dimensiones recibidas. Se tendrá que capturar un valor para cada espacio de la matriz y finalmente imprimir la matriz capturada.
16. Programa que pida las dimensiones de una matriz y declara dos matrices dinámicas de las dimensiones recibidas. Una matriz se captura con datos ingresados por el usuario mientras que la otra es llenada aleatoriamente con números entre 1 y 20. Se van a imprimir ambas matrices tambien con el resultado de la suma de las matrices.

Estructuras dinámicas

17. Programa que declara una estructura llamada usuario que tiene los siguientes datos:
 - Primer nombre
 - Primer apellido
 - Correo

Se capturan datos para cada valor de la estructura y se imprimen.

18. Basado en el ejercicio anterior agrega una variable a la estructura usuario. Esta variable va a ser una variable que apunta a otra estructura llamada dirección. La estructura dirección va a contener los siguientes datos:

- Estado
- País
- Municipio

Finalmente captura datos para ambas estructuras y los imprime en consola.

19. Programa que declare una estructura llamada estudiantes con los siguientes datos:

- Nombre
- Edad
- Calificación

Declaramos de una estructura del tamaño de 20 donde se va a capturar nombre, edad y calificación. Si calificación es mayor o igual que 7 es aprobatoria todo lo demás será reprobatoria. Se tendrá que imprimir lo siguiente:

- Total de aprobados
- Total de reprobados
- Promedio de calificación
- Promedio de edad

20. Programa que declare una estructura dinámica llamada banco. Se tendrá que capturar el numero que se van a capturar banco y se declarara una estructura array del tamaño capturado. Ya se capturarán todos los datos y se imprimen al final.

La estructura llamada banco contendrá las siguientes variables:

- Dinero
- Id_banco
- Datos

Donde datos va a ser otra estructura llamada información que contiene las siguientes variables:

- Nombre
- Edad
- Dir_persona

Donde Dir_persona es otra estructura llamada Dirección que contiene las siguientes variables:

- Numero de calle
- Nombre de calle
- Municipio
- Estado
- País

21. Programa que declara y captura una estructura llamada información que contiene los siguientes datos:

- Nombre
- Edad
- Dir_persona

Donde Dir_persona es una estructura dinámica llamada Dirección que contiene las siguientes variables:

- Numero de calle
- Nombre de calle
- Municipio
- Estado
- País

Antes de capturar los datos de la estructura Dirección se tendrá que preguntar cuántas direcciones tiene la persona y se declara una estructura array del tamaño ingresado. Finalmente se imprime los datos capturados con las direcciones ingresados.

Punteros a funciones

22. Programa que reciba dos números de tipo int. Una vez recibido los valores se imprimen su valor junto. Después se manda la dirección de los números a una función donde se intercambia los valores de los números. Finalmente se imprimen el valor de los números para validar que si fueron cambiados.
23. Programa que declare dos matrices dinámicas con punteros. Se tendrá que pedir el tamaño de las filas y columnas, aquí va a ser el mismo tamaño las filas y columnas. Una vez teniendo esto se tendrá que reservar memoria de las columnas y filas del tamaño introducido. Llenamos las matrices con números pedidos por medio de cin. Finalmente mandamos las matrices a una función con la ayuda de punteros donde se va a imprimir las matrices y se borran de la memoria.
24. Basado en el ejercicio anterior agrega la funcionalidad de:
- Función que imprima el resultado de la suma de las matrices
 - Función que imprima el resultado de la resta de las matrices
25. Basado en el ejercicio anterior agrega la siguiente funcionalidad:
- Captura de tamaños de filas y columnas distintas en las dos matrices
 - Función que imprima el resultado de multiplicar matrices
 - Si las dimensiones de ambas matrices son iguales se hacen las funciones de sumar y restar
 - Si las dimensiones cumplen con las dimensiones de multiplicar matriz se hace la multiplicación. (Columnas primera matriz= filas segundas matriz y filas primeras matriz=Columnas segunda matriz).
 - Si no se cumple con ninguna de las dos condiciones solo se tendrá que imprimir ambas matrices.
26. Programa que declare una estructura dinámica llamada Estudiante que contiene los siguientes datos
- Nombre
 - Edad

- Promedio
- Nombre de materia
- Nombre de maestra/o
- Calificación

Se tendrá que pedir al momento de correr el programa cuantos estudiantes desea capturar y se declara la estructura array de Estudiante de ese tamaño.

Una vez declarada se mandará la estructura array a una función llamada `capturaDatos()` de tipo void. Esta función va a capturar todos los datos pedidos.

27. Basado en el ejercicio anterior agrega las siguientes funciones:

- Función que recibe la estructura principal e imprime todos los datos
- Función que recibe la estructura principal e imprime el promedio, total de aprobados y total de reprobados (Recuerde para aprobar tiene que ser 7 o mayor).
- Función que busca y imprima todos los datos por el nombre de la materia.
- Función que busca y imprima todos los datos por el nombre del maestro.

Manejo de argumentos en la función main

28. Programa que lea y imprima todos los argumentos pasados por el main (Recuerda utilizar la terminal para poder pasar argumentos).

29. Programa que lea los argumentos pasados al main y imprima el resultado de la suma de los argumentos. Validar que si hace la suma bien si les pasas algo que no sea un dígito por el argumento.

30. Programa que lea los argumentos pasados al main y imprima el promedio de los números junto con el resultado de multiplicar los números. Validar que si realiza las operaciones bien si le pasas algo que no sea un entero por argumento.\

31. Programa que lea los argumentos pasados al main. El primer argumento tiene que ser "Multiplicacion", "Division", "Resta" o "Suma". Los siguientes argumentos tiene que ser números donde según el primer argumento pasado se realiza la operación a los siguientes números.
32. Programa que lea los argumentos pasados al main y los imprima ya invertidos los argumentos.

Archivos en C++

Flujo de entrada/salida (librería fstream)

33. Programa que lea lo escrito de un archivo de texto llamado datos.txt (recuerda crear un archivo.txt con información antes de correr este programa). No se te olvide cerrar cualquier stream que fue abierto.
34. Programa que permite escribir a un archivo nuevo llamado prueba.txt con lo que es ingresado por el usuario.
35. Programa que tenga un menú con 3 opciones:
- Insertar
 - Leer
 - Salir

En el de insertar va a insertar nombre de frutas a un archivo llamado frutas.txt, en la opción de leer va a leer todos los datos del archivo y en el de salir se sale del código.

36. Programa que tenga una estructura llamada Jugadores con los siguientes datos:
- Id

- Nombre
- Salario

Este programa va a tener un menú con 5 opciones:

- Insertar
- Leer
- Promedio
- Total
- Salir

El de insertar va a capturar los datos de la estructura y los va a insertar al archivo de texto llamado jugadores.txt, el de leer va a leer todos los datos en el archivo de texto, el de promedio va a sacar el promedio de los salarios, el de total va a imprimir la suma total de los salarios y la opción de salir va a salirse del programa.

37. Basado en el programa anterior agrega las siguientes funcionalidades:

- Función de actualizar datos. Se tiene que dar el numero de id y si existe algo permite actualizar los datos. Si no se encuentra el id que imprima un mensaje de usuario no existe.
- Función de buscar por nombre donde se va a ingresar el nombre y se imprime todos los datos de los usuarios que tenga el nombre igual

38. Programa que tenga un menú con las siguientes opciones:

- Leer archivo binario
- Ingresar datos en archivo binario
- Salir del programa

El archivo binario tendrá nombre de "binario". No se le olvida validar que si sea un archivo binario.

Conceptos de programación orientada a objetos

Definición de clase, atributo, método y objeto

39. Programa que declara una clase llamada Estudiante que tendrá los siguientes atributos públicos que se podrá cambiar los valores desde el main:

- Nombre
- Edad
- Id
- Calificación

Esta clase tendrá un método llamado imprimir que imprime los datos de la clase. En el main se declara un objeto y se inicializa los datos según lo que el usuario ingresa. Finalmente llama el método de imprimir.

40. Basado en el programa anterior cambia los atributos de públicos a privados donde ya se tendrá que hacer una clase de construcción donde se pasan en los parámetros todos los valores a guardar. Se van a crear dos constructores:

- Primer constructor donde se pasan valores para las cuatro variables
- Segundo constructor donde se pasan valores para todas las variables menos la de calificación que se pondrá por default un cero.

Crea dos objetos de esta clase donde cada uno utiliza un diferente constructor y finalmente utiliza el método de imprimir para imprimir ambos objetos de la clase.

41. Programa que calcula varias estadísticas sobre enfermos sobre la población en un intervalo. Primero implementa una clase llamada Estadística que contendrá los siguientes atributos privados:

- Confirmados
- Pruebas
- Población
- Hospitalizados
- Capacidad de hospitales

Se deberá tener un constructor donde se pasan los datos de la clase para cada atributo anteriormente mencionado. Después se tendrá que implementar los siguientes métodos públicos:

- Método que imprime los atributos privados de la clase
- Método que imprime el porcentaje de la población que están confirmados como enfermo.
- Método que imprime el porcentaje de los hospitalizados de los confirmados
- Método que imprime el porcentaje de hospitalizados de la población.
- Método que imprime el porcentaje de pruebas que fueron confirmados.
- Método que imprime el porcentaje de capacidad de hospitales utilizados al momento.

En el main lo que tendrá que hacer es llenar los datos con el constructor y llamar a los métodos implementados.

42. Programa que contenga una clase llamada inversión que contendrá como dato privado una variable entera llamada dinero. La clase tendrá un constructor donde se le pasará la variable del dinero a la clase. El programa tiene la finalidad de encontrar cuanto dinero te regresa cada diferente tipo de inversión. Se tendrá que crear un método para calcular el porcentaje de dinero que regresa en intervalos de 5 años hasta 30 años para cada tipo de inversión. Las inversiones son las siguientes:

Nombre:	Porcentaje:
Retiro	1.6%
Bonos	1.1%
Fondos de inversión	1.3%
Seguro de vida	1.2%

43. La Feria Nacional de San Marcos quiere vender boletos para un evento y va a ocupar un programa como auxiliar para la venta de dichos boletos. El programa va a contener una clase llamada boletos. Debido a que solo se pide un evento ahorita se estará haciendo un constructor por defecto. Esta clase tendrá los siguientes atributos privados:

- Vip
- General
- Dinero

Se tendrá que crear un menú con las siguientes opciones:

- Compra de boletos general. En esta se pedirá cuantos boletos se quiere comprar y comprobar si existen los suficientes boletos. Una vez comprobado que si existen los suficientes boletos se compran los boletos y se reduce los boletos del general.
- Comprar boleto vip. Esta opción es lo mismo que la anterior solo que será para los boletos vip.
- Dinero generado. En esta opción se imprime el dinero total generado, el porcentaje que ha generado cada tipo de boleto y el total de dinero que ha generado cada tipo de boleto.
- Estadística extra. En esta opción se va a tener que publicar promedio de boletos comprados en cada compra para cada tipo de boleto, promedio de dinero gastado en cada compra para cada tipo de boleto, porcentaje de boletos comprados para cada tipo, porcentaje de boletos en venta para cada tipo y finalmente numero de boletos vendidos. Se puede agregar atributos privados a la clase para cumplir con esta opción.
- Salir. Esta opción se va a salir del programa.

Tipo	Precio	Cantidad
General	\$200	500
Vip	\$400	100

44. Basado en el programa anterior se va a agregar otro evento al que se va a vender boletos. Se va a utilizar la misma clase para vender estos boletos. Entonces se tendrá que hacer un constructor donde se pasan los precios y cantidades de los boletos. Los precios y cantidad del segundo evento son como sigue:

Tipo	Precio	Cantidad
General	\$100	100
Vip	\$150	20

Se tendrá que agregar los siguientes métodos:

- Un método que regresa el dinero total generado.
- Otro método donde se pasará la clase y utilizando el puntero this se hace una comparación de dinero generado y imprime cual evento fue ha generado más dinero.

Tendrá que agregarse otro menú donde se elijará entre los dos eventos, otra opción de comparar el dinero que ha generado ambos eventos y finalmente se

moverá la opción de salirse del programa al nuevo menú. Ponle el nombre que usted desea para cada evento.

45. Programa que contenga 3 clases para las siguientes figuras geométricas:

- Rectángulo
- Circulo
- Triangulo

Las 3 clases deberá tener un constructor donde se pasará los parámetros de para sacar el perímetro. Cada clase tendrá un método que regresa el perímetro de las figuras y otro método que utiliza el puntero this para la comparación de dos figuras en donde se pasará un objeto de la clase. Se tendrá que capturar el perímetro para dos figuras para cada clase. Últimamente haz la comparación para ver cual tiene más grande el perímetro.

46. Programa que tenga una clase llamada Números. Tendrá un constructor por default que genera 10 números aleatoriamente entre 1 y 100. Se va a tener que declarar dos diferentes objetos de la clase. Esta clase tendrá que implementar métodos con las siguientes funcionalidades:

- Regresar el numero mas alto de los números generados
- Regresar el número más pequeño de los números generados
- Regresar la suma total de todos los números

Se tendrá que implementar las siguientes funciones en los métodos utilizando el puntero this:

- Imprimir el número más grande generado entre dos objetos de la clase.
- Imprimir el número más pequeño generado entre dos objetos de la clase.
- Imprimir el resultado de la suma de los dos objetos de la clase.
- Imprimir el resultado de la resta de los dos objetos de la clase.

47. Programa que se va a utilizar para calcular los financiamientos de un antro. Se va a declarar una estructura donde viene los siguientes datos:

- Nombre
- Precio de operación por día
- Dinero generado por día
- Costo de construcción

Se tendrá que capturar los datos para cada dato y finalmente se preguntara calcular dinero que se obtiene después de días que será capturado por el usuario.

48. Programa que va a ser utilizado para una frutería donde se van a vender las siguientes frutas:

Nombre	Precio	Cantidad
Manzana	\$10	20
Plátano	\$14	10
Aguacate	\$20	15
Kiwi	\$12	12

Se tendrá que crear una estructura que permita llenarla con los datos proporcionados. Finalmente, un menú que te permita comprar la fruta que quiera el usuario validando que si exista la cantidad que se le pida y muestra con que cantidad de frutas se sigue contando para cada fruta.

49. Programa donde se puede hacer cálculos de dos ecuaciones que contiene números imaginarios. Se tendrá que capturar las dos ecuaciones utilizando una estructura y finalmente tendrás que darle la opción al usuario de elegir una de las siguientes operaciones:

- Suma
- Resta
- Multiplicación
- División

Al imprimir el resultado toma en cuenta si es positivo y negativo al momento de imprimirlo (si el numero imaginario es positivo que se muestra con el + y si es negativo con el -)

Sobrecarga de funciones y operadores

50. Programa donde se crea una clase llamada imprimirdatos donde esta clase va tener 3 funciones con el mismo nombre de print. Cada función deberá ser sobrecargada con un diferente parámetro para imprimir donde los parámetros van a ser de los siguientes tipos:

- Int
- Float
- Double

En el main manda a llamar las 3 funciones con los 3 diferentes tipos de variables para probar la sobrecarga de funciones.

51. Programa donde se sobrecarga una función de tipo void llamada resultado donde estará sobrecargada 4 veces.

- En la primera sobrecarga solo se pasará un int y se imprime el int multiplicado por dos
- En la segunda sobrecarga se pasará un double y se imprime la multiplicación de este número por 3.5
- En la tercera sobrecarga se pasará dos int y se imprime la multiplicación de ambos
- En la cuarta sobrecarga se pasará dos double y se imprime la multiplicación de ambas.

52. Programa donde se declara una clase llamada caja. Esta clase tendrá una función constructora donde se le pasa la altura, largura y ancho de la caja y una función donde imprime el volumen de esta caja. Luego se sobrecarga el

operador de suma para permitir la suma de dos cajas. Finalmente, en el main declara dos cajas y súmalas, el resultado los mandas a una tercera caja y finalmente imprime el volumen de las tres cajas.

53. Basado en el programa anterior agrega la sobrecarga de las resta y multiplicación de las cajas. Declara otras dos cajas donde se mostrará los resultados de la resta y multiplicación.

54. Programa que contenga una clase llamada compleja esta clase tendrá dos variables enteras una llamada real y imaginaria. Esta clase tendrá un constructor, una función de imprimir ecuación y una sobre carga de sumar donde se realizará la suma del valor complejo.

55. Basado en el programa anterior agregarle las siguientes funcionalidades con la sobrecarga de operadores:

- Resta
- Multiplicación
- División

Agregar otras tres cajas para mostrar la funcionalidad de esta.

56. Programa que tenga una clase llamada tiempo que contenga las siguientes variables enteras:

- Horas
- Minutos
- Segundos

La clase tendrá las siguientes funcionalidades:

- Constructor
- Sobrecarga de operadores para hacer la suma
- Imprimir datos de la clase

Declarar dos instancias de la clase sumarlas y igualarlas a una tercera instancia. Imprimir las tres instancias.

57. Programa que tenga una clase llamada que tenga las siguientes variables enteras:

- Dia
- Mes
- Año

Esta clase tendrá un constructor donde se pasarán las tres variables y tendrá una sobrecarga de operadores de << para la impresión de datos de la clase. Deberá tener el formato de mes/día/año.

Repositorio de Github con solución de todos los ejemplos:

<https://github.com/icryeverytime/Programacion-II-Ejemplos>