

 71901020		Fundamentos de Programación		901
		MULTIPLES GRADOS		
Febrero - 2020 Original	Duración: 120 min.	EXAMEN: Tipo B Mixto	Nacional - U.E. 1º Cuatrimestre	
[V%H1e] Material: Ninguno			Hoja 1 de 3	

FUNDAMENTOS DE PROGRAMACION	MODELO 2	CONV. ORD. 19/20
GRADO DE INFORMÁTICA y GRADO DE TECNOLOGÍAS DE LA INFORMACIÓN-ETSII-UNED		
MATERIAL AUTORIZADO: NINGUNO		DURACIÓN: 2 HORAS
INSTRUCCIONES:	1) El test debe ser contestado en la hoja de marcas. Sólo una de las cuatro respuestas posibles de cada pregunta es correcta. 2) El test es eliminatorio y aporta un 30% de la nota final. Son necesarias 8 preguntas correctas (6 con las prácticas aprobadas) para que se corrija el ejercicio. 3) Cada respuesta correcta 1 pto. Respuesta incorrecta, doble o en blanco: 0 pto. 4) Puede quedarse, si lo desea, con esta hoja de examen.	

CUESTIONES DE TEST

1. Dado el siguiente código C++:

```
X Y = {1, 2};
```

- A. X no es un Tipo
- B. Y es un registro
- C. Y no es un Tipo
- D. X e Y valen {1, 2}

2. El siguiente código C++:

```
int funcion(int x) {
    return x*3;
}
void procedimiento(int &x, int y) {
    x = y-2;
}
int main()
{
    int x, y;
    x = 2;
    y = 5;
    procedimiento(x, funcion(y));
    printf("%d; %d", x, y);
}
```

- A. Imprime 2; 5
- B. Imprime 13; 3
- C. Imprime 13; 5
- D. Imprime 2; 3

3.- Dada la siguiente declaración de tipos en código C++:

```
typedef int Vector_Valor[10];

typedef struct miregistro{
    int valor_indice;
    Vector_Valor valores;

    void subir_indice();
};
```

Una sentencia correcta en la implementación del subprograma subir_indice sería:

- A. miregistro.valores[valor_indice]++;
- B. Vector_Valor[valor_indice]++;
- C. valor_indice[indice]++;
- D. valores[valor_indice]++;

		Fundamentos de Programación		901
		MULTIPLES GRADOS		
71901020	Febrero - 2020 Original	Duración: 120 min.	EXAMEN: Tipo B Mixto	Nacional - U.E. 1º Cuatrimestre
[V%H1e] Material: Ninguno				Hoja 2 de 3

4.- Una colección de valores posibles y las operaciones significativas entre ellos se denomina:

- A. Tipo de dato
- B. Variable
- C. Constante
- D. Expresión

5.- En C++, el uso de la sentencia break:

- A. Es obligatoria en todos los casos de un switch
- B. Es opcional en los casos de un switch
- C. Se puede utilizar en cualquier punto de un programa
- D. Es obligatoria en la selección por defecto.

6.- El tratamiento de excepciones permite realizar programas de mayor:

- A. Robustez
- B. Corrección
- C. Eficiencia
- D. Complejidad

7.- El siguiente código C++:

```
int x, y;
x = 2;
y = 7;
do {
    printf("%d ", x-y);
    x = y/3+x;
} while (x%3 == 1);
```

- A. Imprime -6 un número "infinito" de veces
- B. Imprime -5 (una sola vez)
- C. Imprime -5 -3
- D. Imprime -5 un número "infinito" de veces

8.- En C++, los vectores abiertos:

- A. Sólo pueden utilizarse en la definición de funciones
- B. Pueden utilizarse en la definición de funciones y procedimientos
- C. Sólo pueden utilizarse en la definición de procedimientos
- D. No pueden utilizarse ni en la definición de funciones ni de procedimientos

9.- En C++, el uso de la palabra clave const:

- A. Sólo permite declarar constantes
- B. Permite declarar constantes, argumentos y expresiones
- C. Permite declarar constantes y expresiones
- D. Permite declarar constantes y argumentos

 71901020		Fundamentos de Programación		901
		MULTIPLES GRADOS		
Febrero - 2020 Original	Duración: 120 min.	EXAMEN: Tipo B Mixto	Nacional - U.E. 1º Cuatrimestre	
[V%H1e] Material: Ninguno			Hoja 3 de 3	

10.- En el lenguaje C++, el resultado de la siguiente expresión:

`largo < ancho-ajuste && largo*ancho < area`

- A. Es de tipo char
- B. Es de tipo int
- C. Es de tipo float
- D. Es de tipo bool

EJERCICIO DE PROGRAMACIÓN

Realizar un tipo abstracto de datos (TAD) GestionRuta para gestionar una ruta de viaje. Cada ruta tendrá un pueblo origen y un máximo de 5 paradas incluyendo el pueblo destino final. La información de cada ruta será el pueblo origen y entre 1 y 5 tramos, según el número de paradas intermedias. La información de cada tramo será: el pueblo de parada, la duración del tramo y el precio del tramo. Los tramos intermedios serán consecutivos y tendrán como origen la anterior parada o el origen de la ruta. El pueblo del último tramo será el destino final de la ruta.

Las operaciones del TAD son:

- **ListarRuta:** para mostrar todos los datos de la ruta.
- **DatosViaje:** que para un pueblo origen y otro destino devuelve el precio y la duración cuando este viaje es posible dentro de la ruta o un mensaje de viaje no disponible.
- **LargaDistancia:** que devuelve cierto si la ruta tiene una duración total desde el inicio al final mayor de dos horas y falso en caso contrario.