

UNIVERSIDAD INTERAMERICANA DE PANAMÁ  
Facultad de Ingeniería, Arquitectura y Diseño  
Escuela de Ingeniería en Sistema Computacionales  
Programación de Computadoras I – 1Q\_2022

Profesor: Leonardo Esqueda  
Examen

Nombre: Sebastian Bermudez Ospina ID: 100073214

Teoría (Individual):

- 1) Nombre cinco (9) tipos de datos y sus funciones y los límites (según sea el caso) de cada una de ellas.

**R/:**

| Tipo de dato | Función   | Limite  |
|--------------|---|---|
| Int          | Es un tipo de dato que almacena una cantidad entera ya sea positiva o negativa no decimal.      | 2 bytes o una palabra (varía según compilador).   |
| char         | Este tipo de dato se utiliza para representar caracteres individuales.                          | El tipo char requiere sólo un byte de memoria.  |
| float        | Almacena datos de tipo real (decimal) en punto flotante.  | 1 palabra (4 bytes).  |
| string       | Se utiliza para representar y manipular una secuencia de caracteres.                            | Contiene secuencias de puntos de código de 16 bits sin signo (2 bytes) que oscilan entre 0 y 65535. |
| void         | Se utiliza para definir una función que no devuelve ningún valor o declarar punteros genéricos. |   |

- 2) Explique con sus propias palabras la función de una librería a nivel de programación.

**R/:**

**Librería:** La función de una librería en programación es que suelen ser definidas una variedad de funciones, podemos incluir una librería en nuestro programa utilizando rápidamente las instrucciones asignadas allí para que faciliten la programación de dicho programa.

- 3) Explique con sus propias palabras la función de los procesos cíclicos y que prioridad tienen cada uno de ellos

**R/:**

**Procesos Cíclicos o bucles:** Son ciclos repetitivos que permiten repetir instrucciones un número determinado de veces podemos utilizar los siguientes:

- **For:** Se utiliza para repetir instrucciones un determinado número de veces. De entre todos los procesos cíclicos, el FOR se suele utilizar cuando sabemos seguro el número de veces que queremos que se ejecute.
- **While:** Conjunto de instrucciones que se repiten hasta cumplir la condición especificada si su condición es true se ejecutara de manera correcta, pero si es false las instrucciones no serán ejecutadas.
- **Do while:** Sentencia que se ejecuta al menos una vez y es Re ejecutada cada vez que la condición se evalúa verdadera.

La diferencia que hay entre el **while** y **do while** es que **while** evalúa primero la condición y luego ejecuta el proceso en cambio el **do while** ejecuta el proceso y luego evalúa la condición.

- 4) Mencione en que casos usted utilizaría un proceso do-while sobre un proceso for.

**R/:** El caso en se puede utilizar un **do-while** sobre un proceso **for** es cuando queremos repetir las instrucciones cierto número de ocasiones (Por ejemplo 10 veces) se emplea en el recorrido de vectores, matrices y estructuras, entre otros.

- 5) Comente los procesos que se están realizando en el siguiente fragmento de código y al final explique cuál es la función del mismo

**R/:**

```

#include<iostream>
#include<string>
using namespace std;
int main()
{
    string apno;
    float hrtr,tahr,subt,boni,tota;
    cout<<"Calculos de pagos\n\n";
    cout<<"Nombres:\t";cin>>apno;
    cout<<endl<<endl<<"Horas Trabajadas:\t";cin>>hrtr;
    if (hrtr<=0)
        cout<<"No trabajo nada"<<endl;else
    {cout<<"Tarifa por hora:\t";cin>>tahr;
    subt=hrtr*tahr;
    if(hrtr>192)
        boni=subt*0.05;
    else
        boni=subt*0.03;
    tota=subt+boni;
    cout<<"El sub total es:\t"<<subt<<endl;
    cout<<"La bonifiacion es:\t"<<boni<<endl;
    cout<<"El total a pagar es:\t"<<tota<<endl<<endl;
    }cin.ignore(); return 0;
}

```

### Desarrollo punto 5

|   |   |
|---|---|
| <pre> #include&lt;iostream&gt; #include&lt;string&gt; using namespace std; int main() {     string apno;     float hrtr,tahr,subt,boni,tota;     cout&lt;&lt;"Calculos de pagos\n\n";     cout&lt;&lt;"Nombres:\t";cin&gt;&gt;apno;     cout&lt;&lt;endl&lt;&lt;endl&lt;&lt;"Horas Trabajadas:\t";cin&gt;&gt;hrtr;     if (hrtr&lt;=0)         cout&lt;&lt;"No trabajo nada"&lt;&lt;endl;else     {cout&lt;&lt;"Tarifa por hora:\t";cin&gt;&gt;tahr;     subt=hrtr*tahr;     if(hrtr&gt;192)         boni=subt*0.05; </pre> | <p>→ Librería del programa para facilitar dicha programación.</p> <p>→ En esta línea de código están declarando a <b>apno</b> como <b>string</b> (cadena de caracteres).</p> <p>→ Las variables están declaradas como reales.</p> <p>→ Saliendo el anuncio de cálculos de pagos en pantalla.</p> <p>→ En esta línea de código la salida es nombre y la entrada es la variable <b>apno</b> declarada como <b>string</b> donde se ingresa el nombre de la persona.</p> <p>→ En la línea de código se mostrará horas trabajadas que las ingresara la persona y con la ayuda del cin se mostrara en pantalla. Las horas ingresadas.</p> <p>→ El <b>if (si)</b> es un condicional diciendo que si las horas trabajadas son menor o igual a cero mostrar "no trabaja nada " .</p> <p>→ El <b>else(en otro caso)</b> es otro condicional diciéndonos que en otro caso la tarifa por hora utilizando el <b>cin</b> para leer la tahr y luego sacar el subtotal para multiplicar hrtr por tahr.</p> <p>→ Se vuelve a utilizar el condicional <b>if</b> diciendo que si horas trabajadas es mayor a 192 se obtiene una bonificación multiplicando el subtotal por 0.05.</p> |
|---|---|

```

else
    boni=subt*0.03;
    tota=subt+boni;
    cout<<"El sub total es:\t"<<subt<<endl;
    cout<<"La bonifiacion es:\t"<<boni<<endl;
    cout<<"El total a pagar es:\t"<<tota<<endl<<endl;
}cin.ignore(); return 0;
}

```

Se vuelve a utilizar el condicional **else** para decir en otro caso la bonificación se multiplica el subtotal por 0.03 sacando el total sumando subtotal mas bonificación.

Se imprime en pantalla el subtotal, bonificación y el total a pagar ya que en el paso anterior se coloco el dato y ahora los asigna a la línea de código para mostrar al final al momento de compilar..

Finalmente se utiliza el **return** que es el que finaliza la ejecución del programa como tal para ser devuelto en pocas palabras es retroceder.

### Ignore();

Descarta los caracteres del flujo hasta el delimitador dado, inclusive, y luego trae el resto del flujo.

**La funcion del siguiente codigo es calcular el sueldo de los empleados con su nombre y horas trabajadas, ademas si tenia un mayor de 192 horas trabajadas el empleado recibira una bonificacion sumandosele al sueldo para sacar el total a pagar.**

