



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Rodríguez Espino Claudia

Asignatura: Fundamentos de programación

Grupo: 4

No de Práctica(s): 7

Integrante(s): Sánchez Hernández Luis Ricardo

*No. de Equipo de
cómputo empleado* 36

Semestre: 2019-2

Fecha de entrega: 28-03-2019

Observaciones:

CALIFICACIÓN: _____

Objetivos:

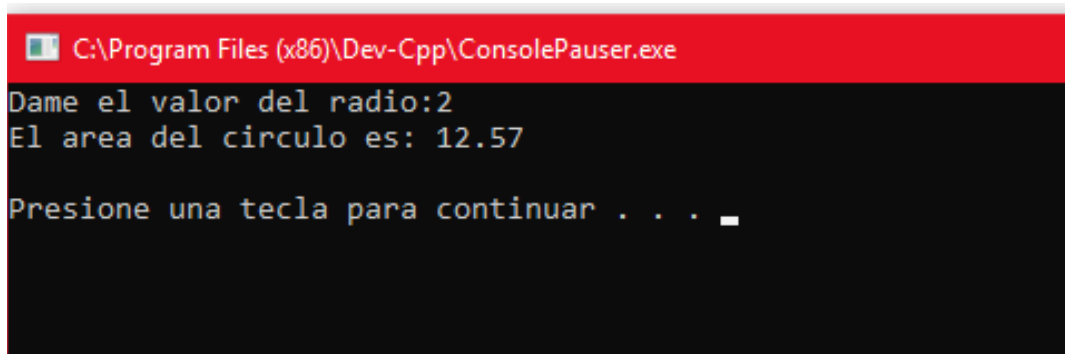
Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

Actividades:

- Crear un programa en lenguaje C que tenga definidas variables de varios tipos, se les asigne valores adecuados (por lectura o asignación directa) y muestre su valor en la salida estándar.
- En un programa en C, asignar valores a variables utilizando expresiones aritméticas; algunas con cambio de tipo(cast)
- Elaborar expresiones relacionales/lógicas en un programa en C y mostrar el resultado de su evaluación.

1.- Programa que calcula el área de un círculo (con cast)

```
area de un círculo con cast.cpp  area de un círculo.cpp
1  #include <stdio.h>
2  int radio, area;
3
4  main()
5  {
6
7      printf("Dame el valor del radio:");
8      scanf("%f", &radio);
9      float area= (float)3.1416*(radio*radio);
10     printf("El area del círculo es: %f\n\n", area);
11 }
```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Dame el valor del radio:2
El area del círculo es: 12.57

Presione una tecla para continuar . . .
```

2.-Programa que resuelve ecuaciones dependiendo del valor asignado por el usuario.

```
voy a intentar hacer el de las ecuaciones.cpp me dices cosas y te digo tu rfc.cpp
1  #include <stdio.h>
2  int x,y;
3
4  main()
5  {
6  printf("Dame el valor de para el que quieres resolver:");
7  scanf("%d", &x);
8  if(x<2)
9  {
10     y=((2*x)-(3*x)+3);
11     printf("la soluci%cn es:%d", 162, y);
12 }
13 else
14 {
15     if(x>2)
16     {
17         y=((3*x*x)+(3*x)-25);
18         printf("la soluci%cn es:%d", 162, y);
19     }
20     else
21     {
22         printf("No tiene soluci%cn", 162);
23     }
24 }
25 }
26
27
28
29
30
31 }
```

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Dame el valor de para el que quieres resolver:2
No tiene soluci%cn
-----
Process exited with return value 0
Press any key to continue . . .
```

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Dame el valor de para el que quieres resolver:-5
la soluci%cn es:8
-----
Process exited with return value 0
Press any key to continue . . .
```

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Dame el valor de para el que quieres resolver:3
la soluci%cn es:11
-----
Process exited with return value 0
Press any key to continue . . .
```

Conclusiones:

En esta práctica observamos la utilidad tanto de la estructuras de secuencia para ahorrarnos el tener que escribir tantas líneas de código que aparte de atrasar el proceso de creación de un programa, terminan haciendo que el código sea mas difícil de leer, lo cual sería quitarle una de sus ventajas al lenguaje mismo que estamos usando, también pudimos comprender la utilidad de la función `cast`, ya que no siempre vamos a poder ingresar valores no enteros como variables para resolver un problema por lo que es una función de gran utilidad para no mostrar resultados que por restricciones del programa sea erróneo.