

# Ejercicios de Programming set del Libro Sebesta

Jimmy Banchon - Rene Balda

12 de febrero de 2014

## 1. Introducción

Los capitulos a realizar son:

1. Ch5: 4 - 7
2. Ch6: 1, 2, 7
3. Ch7: 1 - 6, 9
4. Ch8: 3, 4, 5
5. Ch9: 1, 5

## 2. Preguntas y Respuestas

2.1. Capítulo 5: Nombres, Enlaces y Alcances.

2.1.1. Pregunta 4:

2.1.2. Pregunta 5:

Para la prueba en C, se genera un error.

```
void main( void )
{
    x=21;
    printf(" %d" ,x );
    int x;
    x=42;
    printf(" %d" ,x );
}
```

Errores:

1.-error C2065 x : identificador no declarado  
c:\users\jimmy\documents\visual studio \2011\projects\test\test.c

2.-error C2065 x : identificador no declarado  
c:\users\jimmy\documents\visual studio\2011\projects\test\test.c

Para la prueba en C++, se dio este error al compilar.

```
void main(void)
{
    cout << "Inserta 21:";
    cin >> x;
    int x;
    x=42;
    cout << "X vale:" << x;
    cin.get(); cin.get();
}
```

Errores:

1.-error C2065: 'x' : identificador no declarado  
c:\users\jimmy\documents\visual studio \2011\projects\test\test.cpp

Para la prueba en java, se dio el siguiente error.

```
public class Test {
    static void main(String[] args) {
        x=21;
        int x;
        x=42;
        System.out.println(x);
    }
}
```

Errores:

Exception in thread "main" java.lang.RuntimeException:  
Uncompilable source code - cannot find symbol  
symbol: variable x  
location: class test.NewMain  
at test.NewMain.main(NewMain.java:17)  
Java Result: 1

### 2.1.3. Pregunta 6:

Para la prueba en C++

```
using namespace std;

int main(){
    for (int i = 0 ; i<5; i++){
        cout << i;
    }
    cout << i;
    return 0;
}
```

Errores:

Error test.cpp: En la función int main():  
test.cpp:8:11: error: la búsqueda de nombre de 'i'  
cambió por el nuevo alcance ISO de 'for'  
[-fpermissive]

Para la prueba en Java

```
public class Test{
    public static void main(String args[]){
        for(int i = 0; i< 5 ; i++){
            {
                System.out.println(i);
            }
            System.out.println(i);
        }
    }
}
```

Errores:

Error Test.java:9: error: cannot find symbol  
System.out.println(i);  
symbol: variable i  
location: class test

Para la prueba en C sharp

```
class c5p6
{
    static void Main()
    {
        for(int i = 0 ; i < 5 ; i++)
```

```

        {
            Console.WriteLine(i);
        }
        Console.WriteLine(i);
    }
}

```

Errores:

Error test.cs(11,23): error CS0103: The name 'i' does not exist in the current context

Como podemos ver tanto en lenguaje C++, Java y C sharp no es posible hacer lo que nos pide el problema y nos genera un error, debido a que la variable solo puede ser accedida dentro del bloque for.

2.1.4. Pregunta 7:

2.2. Capítulo 6: Tipos de Dato.

2.2.1. Pregunta 1:

2.2.2. Pregunta 2:

2.2.3. Pregunta 7:

2.3. Capítulo 7: Expressions and Assignment Statements

2.3.1. Pregunta 2:

Traducción al lenguaje C++

```

int fun(int *k); //Prototipo

int main() {
    int i = 10, j = 10, sum1, sum2;
    sum1 = (i / 2) + fun(&i);
    sum2 = fun(&j) + (j / 2);
    printf("sum1 = %d \n ",sum1);
    printf("sum2 = %d ",sum2);
    return 0;
}

int fun(int *k) {
    *k += 4;
    return 3 * (*k) - 1;
}

```

```
}
```

Resultados:

```
sum1 = 46
```

```
sum2 = 48
```

Dado los resultados obtenidos podemos concluir que la asociatividad es de izquierda a derecha al igual que C.

Traducción al lenguaje Java

```
public class Test {
    public static void main(String[] args) {
        Test t = new Test();
        int i = 10, j = 10, sum1, sum2;
        sum1 = (i/2) + t.fun(i);
        sum2 = t.fun(j) + (j/2);
        System.out.println(" Valor 1: "+sum1);
        System.out.println(" Valor 2: "+sum2);
    }

    public int fun(int k){
        k += 4;
        return (3 * (k) - 1);
    }
}
```

Resultados:

```
sum1 = 46
```

```
sum2 = 46
```

Traducción al lenguaje C sharp

```
class Test
{
    static void Main(string[] args)
    {
        Test t = new Program();
        int i = 10, j=10, sum1, sum2;
        sum1 = (i / 2) + t.fun(ref i);
        sum2 = t.fun(ref j) + (j / 2);
        System.Console.WriteLine(" "+sum1);
        System.Console.WriteLine(" "+sum2);
        Console.Read();
    }
    public int fun(ref int k)
```

```

    {
        k = 4 + k;
        return 3*(k) - 1;
    }
}

```

Resultados:

```

sum1 = 46
sum2 = 48

```

Dado los resultados obtenidos podemos concluir que la asociatividad es de izquierda a derecha al igual que C y C++.

#### 2.3.2. Pregunta 4:

Lenguaje Java.

```

public class Test
{
    final static int num=5;
    static int x=5;

    public static void main(String[] args){

        x = fun()+x;
        System.out.println(x);
    }

    static int fun() {
        x = 17;
        return 3;
    }
}

```

Resultados:

```

x = 20

```

Primeramente la funcion fun retorna 3 y actualiza la variable global x con 17 para luego sumarla con 3 asignandole 20 a la variable x. En java el operador + tiene asociatividad de izquierda a derecha.

### 2.3.3. Pregunta 5:

Lenguaje C++

```
int fun();

extern int x = 10;
void main(){

    x = fun()+x;
    a = a+fun();
    printf("\%d", x);
    getch();
}

int fun() {
    x = 17;
    return 3;
}
```

Resultados:

x = 20

Para este lenguaje podemos ver que siempre en la función al llamar fun(), se va a actualizar la variable global a con 17 y luego se le sumara 3 teniendo asi un valor de 20 en dicha variable.

### 2.3.4. Pregunta 6:

Lenguaje C Sharp

```
class Test
{
    static int x = 5;
    static void Main(string [] args)
    {
        x = a +fun();
        Console.WriteLine(x);
        Console.ReadLine();
    }

    static int fun()
    {
        x = 17;
    }
}
```

```
        return 3;
    }
}
```

Resultados: OUTPUT

x = 8

Como podemos ver en C Sharp también se cumple la regla de la asociativad. Es decir la asociatividad va de izquierda a derecha.

2.4. Capítulo 8: Expressions and Assignment Statements

2.5. Capítulo 9: SubProgramas.