

A. Salida por consola y explicación del código inicial
RTA/

El código actual tiene dos métodos funcion activos:
uno para int, uno para double, y otro para ningún tipo
específico. Veamos qué se imprime y por qué:

`System.out.println("char : " + funcion(c));`

Salida: char : Entra a int: 103

Explicación: El carácter 'g' tiene un valor ASCII de 103.
Al pasarlo al método funcion, se convierte automáticamente
en un entero (int), y se invoca el método funcion(int a).

`System.out.println("short : " + funcion(s));`

Salida: long : Entra a double: 9.99999999E8

Explicación: No hay un método funcion(long a). Un valor de tipo
long puede promoverse a double. Por lo tanto, se invoca
funcion(double a).

`System.out.println("integer : " + funcion(i));`

Salida: integer : Entra a int: 51232

Explicación: i es de tipo int, y hay un método funcion(int a)
activo, por lo que se llama directamente a este método.

`System.out.println("double : " + funcion(d));`

Salida: double : Entra a double: 12.4

Explicación: d es un valor de tipo double. El método
funcion(double a) coincide exactamente y se llama.

`System.out.println("float : " + funcion(f));`

Salida: float : Entra a double: 5.65

Explicación: No hay un método funcion(float a) activo.
Un float puede promoverse automáticamente a double,
llamando a funcion(double a).

B. Cambios y sus efectos

1. Activar la función que recibe un short:

```
static String funcion(short a) {  
    return "Entra a short: " + a;  
}
```

Cambio en la salida:

La línea `System.out.println("short : " + funcion(s));` ahora invoca el método `funcion(short a)` en lugar de `funcion(int a)`.

Nueva salida: short : Entra a short: 2
El resto de las líneas permanecen igual.

2. Activar la función que recibe un float:

```
static String funcion(float a) {  
    return "Entra a float: " + a;  
}
```

Cambio en la salida:

La línea `System.out.println("float : " + funcion(f));` ahora llama a `funcion(float a)` en lugar de promover `f` a `double`.

Nueva salida: float : Entra a float: 5.65
El resto de las líneas permanecen igual.

3. Comentar la función que recibe un double y activar la que recibe un float:

```
// static String funcion(double a) {  
//     return "Entra a double: " + a;  
// }  
static String funcion(float a) {  
    return "Entra a float: " + a;  
}
```

Cambio en la salida:

Cualquier valor que promovía a `double` ahora buscará el método `funcion(float a)` si aplica, o se elevará a un método más general.

Salida para long :

Ahora `long` no tiene un método directo ni se puede promover a `float`, por lo que no hay una coincidencia. Esto probablemente cause un

error de compilación.

Salida para float :

Se mantiene como float : Entra a float: 5.65.

4. Comentar todas las funciones, excepto la que recibe un double

```
static String funcion(double a) {  
    return "Entra a double: " + a;  
}
```

Cambio en la salida:

Todos los valores que no tienen una función directa se promoverán a double si es posible.

Salida para char : Se convierte en un double,
e imprime: char : Entra a double: 103.0.

Salida para short : y byte : Se convierte en un double,
e imprime algo como: short : Entra a double: 2.0.

Salida para long : Igual, imprime:
long : Entra a double: 9.99999999E8.

Salida para int : También, imprime:
integer : Entra a double: 51232.0.