

PRUEBA 1 PROGRAMACIÓN
Noviembre 2008
INGENIERÍA SISTEMAS COMUNICACIÓN
Leganés



Universidad
Carlos III de Madrid

LEA ATENTAMENTE ESTAS INSTRUCCIONES ANTES DE COMENZAR LA PRUEBA:

- Rellene todas las hojas a bolígrafo, tanto los datos personales como las respuestas
- No utilice lápiz ni bolígrafo rojo
- No olvide rellenar el NIA y el grupo real al que pertenece
- El tiempo máximo de realización es de 1 hora
- El único material permitido sobre la mesa es la hoja de test y un bolígrafo
- Utilice exclusivamente esta hoja de test para las respuestas, use las caras posteriores para contestar si lo necesita. No se recogerá ninguna otra hoja adicional.

NO PASE DE ESTA HOJA, hasta que se le indique

Apellidos	Nombre	
Firma	NIA	Grupo

PARTE 1: CUESTIONES

Pregunta 1 (1 Punto).- Indicar si la siguiente afirmación es cierta, y **explicar** brevemente por qué.

“Utilizando || en lugar de | siempre obtendremos el mismo resultado en una expresión”

Verdadero. Tanto I como II realizan la operación lógica OR, la diferencia entre ambos está en que I evalúa los dos términos de la expresión, mientras que II en el caso de que el primero sea cierto (y por lo tanto el resultado será cierto independientemente del valor del segundo) no evalúa el segundo. Por lo tanto el resultado de la operación lógica no varía, aunque el uso de uno u otro puede tener efectos colaterales distintos como por ejemplo:

```
int a=5;  
boolean b = a>0 || a++>1;
```

Como a es mayor que cero, a++ no se ejecutaría y seguiría valiendo 5, pero si hubiéramos puesto |, a valdría 6.

Pregunta 2 (1 Punto).- Indicar si la siguiente afirmación es cierta, y **explicar** brevemente por qué.

“Si no inicializamos un array de cualquier tipo básico, Java le da un valor por defecto a cada uno de los elementos del array (0 para los tipos numéricos, false para boolean, el carácter vacío para char y null para String)”

Verdadero. Java no da valor por defecto a las variables de tipos básicos, pero sí a los elementos de un array.

Pregunta 3 (1 Punto).- Indicar si la siguiente afirmación es cierta, y **explicar** brevemente por qué.

“Podemos mezclar variables de distintos tipos en una expresión aritmética y Java decidirá automáticamente el tipo del resultado”

Falso. En una expresión aritmética se pueden mezclar todos los tipos de datos **numéricos**, e incluso char, pero no boolean.

En caso de que los tipos sean numéricos o char, el tipo del resultado será el del operando de mayor rango, excepto en el caso en el que se opere exclusivamente con char, byte o short en cuyo caso el resultado será de tipo int

Pregunta 4 (1 Punto).- Encontrar y **explicar** los 4 errores de compilación que aparecen en el siguiente código Java. ¿Cómo los resolvería?

```
public class ProblemaCuatro {
    public static void main(String[] args) {
        int x, y;
        byte z = 128;
        short w, v = 125;
        double s = 3.5F, x = 0.0;
        char t = 148;

        x = 4;
        z = x + y;

        s = x + v;

        System.out.println("El valor de s es: " +(x+v));
        System.out.println("La letra siguiente a la t es: " ++g);
    }
}
```

Error 1: Se da valor 128 a z, que es tipo byte. Se sale de rango, por lo que habría que hacer un casting (byte z = (byte) 128; el resultado no sería el esperado)

Error 2: Se declara 2 veces una variable de nombre x, una como int y otra como double. Habría que cambiar el nombre de alguna de las dos.

Error 3: Se hace z = x + y, pero no se ha dado valor a y anteriormente. Se tendría que haber inicializado y antes. Además, x e y son enteros, y la suma, también es entero, por lo que habría que haber hecho casting: z = (byte) (x+y)

Error 4: g no está definida

Pregunta 5 (1 Punto).- **Explicar** el resultado de las siguientes operaciones:

```
public class ProblemaCinco {
    public static void main(String[] args) {
        int i,j;
        for(i=0; i<=1; i++) {
            for(j=0; j<2; j++) {
                System.out.println(j+i);
            }
        }
    }
}
```

El programa tiene dos bucles for anidados, el primero se ejecuta entre 0 y 1 y el segundo también, cada vez que se ejecuta el segundo bucle se imprime la suma de los dos índices. Por lo tanto el resultado será:

i=0, j=0 → imprime 0

i=0, j=1 → imprime 1

i=1, j=0 → imprime 1

i=1, j=1 → imprime 2

Pregunta 6 (1 Punto).- Dada las siguientes declaraciones, **explicar** cuáles son correctas y cuáles no. En las incorrectas, ¿hay alguna forma de corregirlas sin cambiar el tipo de las variables? En caso afirmativo, diga cómo se haría.

- a) `int a= 300; byte b = a;`
- b) `int a1 = 90; char b1=a1;`
- c) `float a2= 4; double b2=a2/0;`
- d) `char a3 = 'g', b3 = a3+1;`
- e) `long a4 = 5; int b4=a4/3;`

- a) Incorrecta, intentamos asignar un dato de tipo `int` a una variable de tipo `byte`. Se resolvería con un casting: `byte b = (byte) a`, aunque al superar 300 el rango de `byte` el resultado almacenado en `b` no será el deseado.
 - b) Incorrecta, intentamos asignar un dato de tipo `int` a una variable de tipo `char`. Se resolvería también con un casting: `char b1 = (char) a1`
 - c) Correcta, asignamos un `int` a un `float` y luego a un `double` que valdrá `Infinity`
 - d) Incorrecta, al sumar 1 a un `char`, se convierte automáticamente en un `int`, y pasa lo mismo que en b). Se resuelve con un casting: `b3 = (char) (a3+1)`
 - e) Incorrecta, asignamos un `long` a una variable de tipo `int`. Se resuelve con un casting: `int b4= (int) a4/3`
-

PARTE 2: PROBLEMAS

Problema 1 (2 Puntos).- En la asignatura de programación del grado de ingeniería en sistemas de comunicación hay 3 grupos de alumnos, que tienen respectivamente 29, 37 y 25 alumnos. Crear un array de `String` para representar esta estructura de forma que cada posición del array pueda contener el nombre y apellidos de un alumno. Crear un alumno en cada uno de los grupos.

```
String[][] programacion = new String[3][];
programacion[0] = new String[29];
programacion[1] = new String[37];
programacion[2] = new String[25];

// Inicializamos el primer alumno de cada grupo
programacion[0][0] = "Perico Robles";
programacion[1][0] = "Marujita Diaz";
programacion[2][0] = "Fulanita Gonzalez";
```

Problema 2 (2 Puntos).- Crear un array de 1001 posiciones de tipo `int` y rellenarlo de la siguiente manera:

- Las casillas pares deberán tener como valor el número de casillas que quedan hasta el final del array, es decir, la 0 valdrá 1000, la 2 valdrá 998, la 4 valdrá 996, y así sucesivamente.
- Las casillas impares deberán tener como valor el doble del valor de la casilla anterior, es decir, la 1 valdrá 2000, la 3 valdrá 1996, la 5 valdrá 1992 y así sucesivamente.

```
int[] miArray = new int[1001];
for(int i=0, j=1000; i<miArray.length; i++, j--){
    if(i%2 == 0) {
        miArray[i] = j;
    } else {
        miArray[i] = 2 * miArray[i-1];
    }
}
```