

Lesson 3. Introduction to Programming in C

Exercises

Exercise 1. Write a C program that prints on the screen a welcome message and asks the user's birth year, which must be stored in a variable named `date`. The program must calculate the current age of the user and print on the screen `You were born in <date>. You are <age> this year, being date and age the read and the calculated value, respectively`. You are recommended to create the program incrementally, and to check compilation errors after developing each program section.

Exercise 2. Write a C program that declares three variables (`a`, `b`, `c`), assigns the values 5, 7, 9 (respectively), and calculates the sum of them. Modify the program to read the values on the keyboard.

Exercise 3. Write a program that calculates the interest gained after a bank deposit. The program must ask the initial amount and the interest rate.

Exercise 4. Write a program that calculates the area of a right-angled triangle. The program must ask the height and the length of the base. The output of the program must be: `Right-angled triangle of height __ and base __. Area __.` (Blank spaces are filled with the actual values).

(NOTE: $\text{Area} = (\text{base} * \text{height}) / 2$)

Extend the program to calculate the area of a circumference.

Exercise 5. Mark variable declarations, assignment operations, operators, and expressions of the following code. Which is the result of the evaluation of each expression?

```
int i1, i2, i3;
float r1, r2, r3;

i1=12;
i2=5;
r1=12.0;
r2 =5;
r3= r1/r2;
r3=i1/i2;
i3=i1/r2;
```

Exercise 6. Write a program to calculate the remainder of the division of two integers. Use the module operator (`%`).

Exercise 7. Write a program that asks for a time value (in seconds) and transforms it into minutes and seconds.



Exercise 8. Write a program that asks for the user's age and test if it is larger than 21 years. If the read value is larger, the program must print 1; otherwise, the program must print 0.

Exercise 9. Write a program that declares two variables (*a*, *b*), assigns them two values read from the keyboard, and swap their values.

Exercise 10. Write a C program to convert between *peseta* and *euro* currencies (1 euro = 166.386 pesetas). The program must read the amount in *pesetas* from the keyboard and print on the screen the conversion to *euros*. Extend the program to convert the resulting *euros* to *dollars* as well (1 euro = 1.31 dollars).

Exercise 11. Are the following statements right?

```
i = i + 1
printf("%d", 4+20);
printf("sum = var_one + var_two = %d + %d", var_one, var_two, sum);
```

Exercise 12. Write a program that ask for a temperature in the Fahrenheit scale and transforms it into the Celsius scale.

(NOTE: Celsius = (Fahrenheit – 32) * 5 / 9)

Exercise 13. Write a C program that prints on the screen a welcome message and asks the user's name. The program must print a personalized greetings message.

Exercise 14. Write a program that declares an integer variable and assigns it a value. The program must show the value of the variable and its memory address.

Exercise 15. Which is the value of *r* after the following code snippet?

```
float n=6.0, m=2.0, r;
r = 25.0 + 120 * n / m;
```

Exercise 16. Find the errors of the following program:

```
int main(void) {
    foat radius, circum;
    printf(type the radius);
    scanf("%f", &radius);
    circum = 2 * PI * radius
    printf("%f", circum);
    return 0;
```

Exercise 17. Write a C program to assign the value 4/0 to an integer variable. What happens when the program is compiled? What happens when the program is executed?

Exercise 18. Write a C program that reads two integer values from the keyboard (*x*, *y*) and prints on the screen *x/y* and *x%y*. Run the program several times with different inputs. What happens when the variable *y* gets the value 0?



Exercise 19. Write a C program that reads three integer values from the keyboard a , b , c and print on the screen 1 if $(a > b > c)$, 0 otherwise.

Exercise 20. Write a C program that declares two variables, x (integer) and y (float) and directly assign them the values 6 and 2.0. The program must print on the screen the results of the expressions below:

```
x * y
3 * x * y
3 * x / y
3 * x % y
```

What happens when the program is compiled? How would you fix this error?

Exercise 21. Write a C program with the code below and compile it. Is there any error?

```
#include <stdio.h>

int main(void){
    int a, b;
    char string[8];
    int c;
    a=7; b=14; c=128;

    printf ("Type a word: ");
    scanf("%[^\\n]", string);

    printf("String is %s\\n", string);
    printf("Variables values are:\\na=%d\\nb=%d\\nc=%d\\n", a, b, c);

    system("pause");
    return 0;
}
```

Run the program by using Hello as input. What happens? Run the program again by entering Supercalifragilisticexpialidocious. What happens?

Exercise 22. Write a C program with the code below and compile it. Study the output of the program as a result of the use of different format modifiers.

```
#include <stdio.h>

int main (void) {
    printf("Type formats in printf \\n\\n");
    printf("-----\\n");
    printf("Floating point values \\n");
    printf("Float literal 1234.12345678\\n");
    printf("> 8 spaces, 1 decimal value 8.1f: [%8.1f] \\n", 1234.12345678);
    printf("> 8 spaces, 3 decimal values 8.3f: [%8.3f] \\n", 1234.12345678);
    printf("> 2 spaces, 3 decimal values 2.3f: [%2.3f] \\n", 1234.12345678);
    printf("> 12 spaces,6 decimal values 12.6f: [%12.6f] \\n", 1234.12345678);
    printf("> 20 spaces,6 decimal values 20.6f: [%20.6f] \\n", 1234.12345678);
    printf("> same as before, left align -20.6f: [%-20.6f] \\n", 1234.12345678);
    printf("\\n-----\\n");
    printf("Strings \\n");
```



```
printf("String [hello madrid] \n");  
printf("> Direct print: [hello madrid] \n");  
printf("> With placeholder: [%s] \n", "hello madrid");  
printf("> Using width specifier (20 spaces, 4 letters) 20.4s: [%20.4s] \n",  
      "hello madrid");  
  
printf("\n");  
system ("pause");  
return (0);  
}
```

