

1 ICS 104 - Introduction to Programming in Python and C

1.1 Overview of C - Lab 1

2 Lab Learning Outcomes

- To properly use data types in C.
- To write assignment statements in C.
- To evaluate arithmetic expressions in C.


2.1 Online C Compilers

https://www.onlinegdb.com/online_c_compiler (https://www.onlinegdb.com/online_c_compiler)

<https://www.codechef.com/ide> (<https://www.codechef.com/ide>)

Precedence and Associativity Rules

In C, mathematical expressions are evaluated according to the following precedence and associativity rules:

		Operators	Order of Evaluation of operands with same precedence (Associativity)
Higher Priority  Low Priority	1	<i>(expression)</i> and function calls	Left to right
	2	unary +, unary – Type cast: (type)	Right to left
	3	*, /, %	Left to right
	4	binary +, binary -	Left to right
	5	=	Right to left

Note: The remainder operator % gives the remainder produced by dividing two integer numbers. Example $5 \% 2 = 1$ and $1 \% 3 = 1$. In C, this operator is not used with double or float values.

1. Standard Mathematical Functions

To do some advanced mathematical calculations, C provides a header file called `math.h` that has different mathematical function prototype definitions. The prototypes of all mathematical functions below, except `abs(x)`, are defined in `math.h`:

C function	Mathematical Notation	Example	Comment
abs(x)	$ x $	abs(-4) = 4	Returns the absolute value of an integer expression. The prototype is defined in stdlib.h
fabs(x)	$ x $	fabs(-7.56) = 7.56	Returns the absolute value of an expression of type double . The prototype is defined in math.h
ceil(x)	$\lceil x \rceil$	ceil(45.1) = 46.0 ceil(-7.9) = -7.0	Returns the smallest integral value (of type double) greater or equal to x
floor(x)	$\lfloor x \rfloor$	floor(12.99) = 12.0 floor(-5.1) = -6.0	Returns the largest integral value (of type double) less than or equal to x
pow(x,y)	x^y	pow(5.0,3) = $5.0^3 = 125.0$	
sqrt(x)	\sqrt{x}	sqrt(4.0) = $\sqrt{4.0} = 2.0$	
log(x)	$\ln x$ (x > 0)	log(2.71828) = 1.0	
log10(x)	$\log_{10} x$ (x > 0)	log ₁₀ (100.0) = 2.0	
exp(x)	e^x	exp(1.0) = 2.171828	
sin(x)	$\sin x$ (x in radians)	sin(1.5708) = 1.0	
cos(x)	$\cos x$ (x in radians)	cos(0.0) = 1.0	
tan(x)	$\tan x$ (x in radians)	tan(0.0) = 0.0	
asin(x)	$\sin^{-1} x$ $x \in [-1,1]$	asin(0) = 0	
acos(x)	$\cos^{-1} x$ $x \in [-1,1]$	acos(0) = 1.570796	
atan(x)	$\tan^{-1} x$ $x \in (-\infty, \infty)$	atan(0) = 0	

3 Example

- Write a program that reads the area of a rectangle (in square cm) and its length (in cm). It then computes and displays the perimeter

of the rectangle in cm.

Analysis: Input: area [in square cm], length [in cm]

Input restrictions: $\text{area} > 0$, $\text{length} > 0$, $\text{area} > \text{length}$

Constants: None

Relevant Formulas: $\text{width} = \text{area} / \text{length}$, $\text{perimeter} = 2 * (\text{length} + \text{width})$

Output: perimeter in cm

Pseudo-code algorithm:

1. Prompt for area[square cm] and length [cm]
2. Input: area, length
3. Compute width:

$\text{Width} = \text{area} / \text{length}$

4. Compute perimeter:

$\text{Perimeter} = 2 * (\text{length} + \text{width})$

5. Output: "Perimeter = ", perimeter, " cm"
6. Stop.

The translation of the above pseudo-code into a C program:

```
#include <stdio.h>
```

```
int main(void) {
```

```
double area, length,width,prmter;    //use meaningful names for variables

printf("Enter area (cm2) and length (cm): ");

scanf("%lf%lf",&area, &length);

width = area / length;

prmter = 2 * (length + width);

printf("perimeter = %.2f cm\n",prmter);

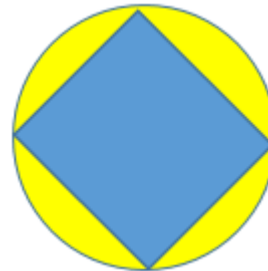
return 0;

}
```

4 Exercises

4.1 Exercise # 1:

A square is inscribed inside a circle. Write an interactive C program that prompts for and reads the radius of the circle in centimeters, it then calculates and prints the side length of the square, the area of the square and the area of the yellow part in the diagram below, each in square centimeters.



- Print each output value with two digits after the decimal point.
- Note: Define the value of π (3.141592) as a constant.
- Following is a sample run:

Enter the radius value [cm]: 4.5

Square side = 6.36 cm

Square area = 40.50 square cm

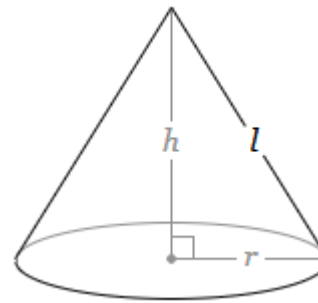
Yellow area = 23.12 square cm

In []:

```
1 // your code for exercise 1
2
3 #include <stdio.h>
4 #include <math.h>
5 #define PI 3.141592
6 int main()
7 {
8     double radius=0;
9
10    printf("Enter the radius value [cm]: ");
11    scanf("%lf", &radius);
12    double squareSide = radius*sqrt(2);
13    double squareArea = pow(squareSide,2);
14    double yellowArea = (PI * radius * radius) - squareArea;
15    printf("Square side = %.2lf cm\n", squareSide);
16    printf("Square area = %.2lf square cm\n", squareArea);
17    printf("Yellow area = %.2lf square cm\n", yellowArea);
18
19
20    return 0;
21 }
22
```

4.2 Exercise # 2:

Write a complete C program that prompts for and reads the height [in cm] and the surface area of a cone [in cm²], it then calculates and displays the volume [in cm³] and the length of the slant side l [in cm].



$$volume = \pi r^2 \frac{h}{3}$$

$$l = \sqrt{h^2 + r^2}$$

$$r = \sqrt{\frac{surfaceArea^2}{\pi(h^2 + 2 surfaceArea)}}$$

Sample program runs:

```
Enter height [cm] and surface area [cm2] of a cone: 5.0 100.0
```

```
Volume = 59.84 cubic cm
```

```
Slant side = 6.04 cm
```

```
Enter height [cm] and surface area [cm2] of a cone: 7.5 240.0
```

```
Volume = 219.27 cubic cm
```

```
Slant side = 9.17 cm
```

In []:

```

1 //your code for exercise 2
2
3 #include <stdio.h>
4 #include <math.h>
5 #define PI 3.141592
6 int main()
7 {
8     double height , surfaceArea;
9     printf("Enter height [cm] and surface area [cm2] of a cone: ");
10    scanf("%lf %lf", &height, &surfaceArea);
11    double radius = sqrt(pow(surfaceArea,2)/(PI *(PI * pow(height,2) + 2*surfaceArea)));
12    double volume = PI * pow(radius,2)*height/3;
13    double slantSide = sqrt(pow(height,2) + pow(radius,2));
14    printf("Volume = %.2lf cubic cm\n" , volume);
15    printf("Slant side = %.2lf cm\n" , slantSide);
16
17    return 0;
18 }
19

```

4.3 Exercise # 3:

Given force F in Newtons, acting between two electrically charged small spheres, the charge of one of the spheres in Coulombs, and the distance between the charges in meters. Write an interactive program to find the charge of the other sphere in Coulombs. Use Coulomb's Law:

$$F = (K * q1 * q2)/r^2$$

Where: K = 8.9875 * 10⁹ Newtons Meter² / Coulombs² ; It is Coulomb's constant. r is the distance between the spheres in meters q1 and q2 are charges of the spheres in Coulombs.

Note: Define K as a constant and give it the value 8.9875e9

Note: In C a number can be written in Scientific notation by using the syntax: mantissa e IntegerExponent or mantissa E IntegerExponent

Example 4.52e4 is 4.52 * 10⁴ and 3.567e-3 is 3.56 * 10⁻³

The format specifier %e or %E is used in printf to output numbers in Scientific notation X.XXXXXXe+XXX or X.XXXXXXe-XXX. This format specifier can also be used in scanf to read a float value. To read a value of type double in scientific notation use %lf or %le [Note:the character in the format specifier is lowercase l not one]

Sample program run:

```
Enter the charges q1 [in Coulombs]: 5.25e-9
Enter the distance between the charges [in meters]: 0.004
Enter the force between the charges [in Newtons]: 1.999438e-2
Charge2 = 6.780000e-009 Coulombs
```

In []:

```
1 //your code for exercise 3
2
3 #include <stdio.h>
4 #include <math.h>
5 #define K 8.9875e9
6 int main()
7 {
8     double q1,q2,distance,force;
9     printf("Enter the charges q1 [in Coulombs]: ");
10    scanf("%le",&q1);
11    printf("Enter the distance between the charges [in meters]: ");
12    scanf("%lf",&distance);
13    printf("Enter the force between the charges [in Newtons]: ");
14    scanf("%le",&force);
15    q2 = (force * pow(distance,2)) / (K * q1);
16    printf("Charge2 = %e Coulombs" , q2 );
17    return 0;
18 }
```