

Computing I – Programming Labs

Problem Set 3 - Functions

Problem 1 (Even Numbers)

Write a program that inputs a series of integers and passes them one at a time to function `isEven`, which uses the modulus operator to determine whether an integer is even. The function should take an integer argument and return `true` if the integer is even and `false` otherwise. It should ask the user to input three integers at a time as shown below.

```
Enter integer 1: 3
3 is an odd integer

Enter integer 2: 2
2 is an even integer

Enter integer 3: 97
97 is an odd integer
```

Problem 2 (Multiples)

Write a function `multiple` that determines for a pair of integers whether the second is a multiple of the first. The function should take two integer arguments and return `true` if the second is a multiple of the first, `false` otherwise. Use this function in a program that inputs a series of pairs of integers, three for each run of the program as shown below.

```
Enter two integers: 4 8
8 is a multiple of 4

Enter two integers: -4 8
8 is a multiple of -4

Enter two integers: -4 -8
-8 is a multiple of -4
```

Problem 3 (Hypotenuse Calculations)

Define a function `hypotenuse` that calculates the hypotenuse of a right triangle when the other two sides are given. The function should take two `double` arguments and return the hypotenuse as a `double` with a precision of one digit after the decimal point. Use this function in a program to determine the hypotenuse as shown below. For each run, the program should calculate the hypotenuse three times as shown below.

```
Enter 2 sides of right triangle: 2.0 3.0
Hypotenuse: 3.6

Enter 2 sides of right triangle: 3.0 4.0
Hypotenuse: 5.0

Enter 2 sides of right triangle: 4.0 5.0
Hypotenuse: 6.4
```

Problem 4 (German Grades)

Write a function germanGrade that inputs a student's number of points (test score) as an integer value and returns

- 1.0 if a student's points are 95–100,
- 1.3 if a student's points are 90–94,
- 1.7 if a student's points are 85–89,
- 2.0 if a student's points are 80–84,
- 2.3 if a student's points are 75–79,
- 2.7 if a student's points are 70–74,
- 3.0 if a student's points are 65–69,
- 3.3 if a student's points are 60–64,
- 3.7 if a student's points are 55–59,
- 4.0 if a student's points are 50–54 and
- 5.0 if a student's points are lower than 50.

The program should allow the input of eleven grades at a time. Your output should resemble the screen shot shown below.

```
Enter the student's test score: 96
96 on the German grade scale is 1.0

Enter the student's test score: 91
91 on the German grade scale is 1.3

Enter the student's test score: 85
85 on the German grade scale is 1.7

Enter the student's test score: 81
81 on the German grade scale is 2.0

Enter the student's test score: 75
75 on the German grade scale is 2.3

Enter the student's test score: 72
72 on the German grade scale is 2.7

Enter the student's test score: 67
67 on the German grade scale is 3.0

Enter the student's test score: 60
60 on the German grade scale is 3.3

Enter the student's test score: 56
56 on the German grade scale is 3.7

Enter the student's test score: 51
51 on the German grade scale is 4.0

Enter the student's test score: 49
49 on the German grade scale is 5.0
```

Problem 5 (Celsius and Fahrenheit Temperatures)

Implement the following integer functions (integer input, integer output):

1. Function `celsius` returns the Celsius equivalent of a Fahrenheit temperature.
2. Function `fahrenheit` returns the Fahrenheit equivalent of a Celsius temperature.

	from Fahrenheit	to Fahrenheit
Celsius	$[^{\circ}\text{C}] = ([^{\circ}\text{F}] - 32) \times \frac{5}{9}$	$[^{\circ}\text{F}] = [^{\circ}\text{C}] \times \frac{9}{5} + 32$
Kelvin	$[K] = ([^{\circ}\text{F}] + 459.67) \times \frac{5}{9}$	$[^{\circ}\text{F}] = [K] \times \frac{9}{5} - 459.67$

(<https://en.wikipedia.org/wiki/Fahrenheit>)

Use these functions to write a program that prints charts showing the Fahrenheit equivalents of all Celsius temperatures from 0 to 100 degrees, and the Celsius equivalents of all Fahrenheit temperatures from 32 to 212 degrees. Print the outputs in a neat tabular format as shown below:

Fahrenheit equivalents of Celsius temperatures:							
Celsius	Fahrenheit	Celsius	Fahrenheit	Celsius	Fahrenheit	Celsius	Fahrenheit
0	32	25	77	50	122	75	167
1	33	26	78	51	123	76	168
2	35	27	80	52	125	77	170
3	37	28	82	53	127	78	172
4	39	29	84	54	129	79	174
5	41	30	86	55	131	80	176
6	42	31	87	56	132	81	177
7	44	32	89	57	134	82	179
8	46	33	91	58	136	83	181
9	48	34	93	59	138	84	183
10	50	35	95	60	140	85	185
11	51	36	96	61	141	86	186
12	53	37	98	62	143	87	188
13	55	38	100	63	145	88	190
14	57	39	102	64	147	89	192
15	59	40	104	65	149	90	194
16	60	41	105	66	150	91	195
17	62	42	107	67	152	92	197
18	64	43	109	68	154	93	199
19	66	44	111	69	156	94	201
20	68	45	113	70	158	95	203
21	69	46	114	71	159	96	204
22	71	47	116	72	161	97	206
23	73	48	118	73	163	98	208
24	75	49	120	74	165	99	210
	100			212			

Celsius equivalents of Fahrenheit temperatures:							
Fahrenheit	Celsius	Fahrenheit	Celsius	Fahrenheit	Celsius	Fahrenheit	Celsius
32	0	77	25	122	50	167	25
33	0	78	25	123	50	168	25
34	1	79	26	124	51	169	26
35	1	80	26	125	51	170	26
36	2	81	27	126	52	171	27
37	2	82	27	127	52	172	27
38	3	83	28	128	53	173	28
39	3	84	28	129	53	174	28
40	4	85	29	130	54	175	29
41	5	86	30	131	55	176	30
42	5	87	30	132	55	177	30
43	6	88	31	133	56	178	31
44	6	89	31	134	56	179	31
45	7	90	32	135	57	180	32
46	7	91	32	136	57	181	32
47	8	92	33	137	58	182	33
48	8	93	33	138	58	183	33
49	9	94	34	139	59	184	34
50	10	95	35	140	60	185	35
51	10	96	35	141	60	186	35
52	11	97	36	142	61	187	36
53	11	98	36	143	61	188	36
54	12	99	37	144	62	189	37
55	12	100	37	145	62	190	37
56	13	101	38	146	63	191	38
57	13	102	38	147	63	192	38
58	14	103	39	148	64	193	39
59	15	104	40	149	65	194	39
60	15	105	40	150	65	195	40
61	16	106	41	151	66	196	41
62	16	107	41	152	66	197	41
63	17	108	42	153	67	198	42
64	17	109	42	154	67	199	42
65	18	110	43	155	68	200	43
66	18	111	43	156	68	201	43
67	19	112	44	157	69	202	44
68	20	113	45	158	70	203	45
69	20	114	45	159	70	204	45
70	21	115	46	160	71	205	46
71	21	116	46	161	71	206	46
72	22	117	47	162	72	207	47
73	22	118	47	163	72	208	47
74	23	119	48	164	73	209	48
75	23	120	48	165	73	210	48
76	24	121	49	166	74	211	49
				212	100		

Note

1. `cin; cin.get(); getline()`

`cin`: can input any data type, especially use in numbers.

`cin.get()`: `Cin.get(ch)`, `ch = cin.get()` → use to enter character

		0	0	'\n'
--	--	---	---	------

↑
`Cin (read from`

↳ `Cin (stop at the new line character)`

↳ `where the user entered)`

↳ `Cin.get() (start from where last time
stopped)`

`getline()`: `getline(Cin, inputLine)`, specifically use in string. Read space as well while `cin` ignore space

`Cin.ignore()`: ignore one character often '\n'

`Cin.ignore(20, '\n')`

↓
Skip 20 characters or encounter new line character.

2. '\n' → character literal

"\n" → string literal ⇒ {'\n', '\0'}] both shift the pointer to another line

`endl` → insert a new line and flushes the stream

'\0' → mark the ending of the string, by default string contains '\0' at the end

