

# CS 110 - Programming I

## Homework 2

### General Instructions

- Every problem needs to be solved in a separate .cpp file
- The name of the cpp file needs to be the number of the problem. For example, the file for the solution to problem 2 should be named 2.cpp
- The first two lines of every one of your solutions should be a comment with your full name and your email address, for example:

```
// Pablo Gallastegui  
// pablo.gallastegui@brescia.edu
```

- All the solutions need to be compressed (zipped up) in a single file named as the left half (before the @ symbol) of your email address, for example:

```
pablo.gallastegui.zip
```

- The compressed folder has to be submitted using moodle, before the due date and time.
- After every homework is due, a few students may randomly be selected to explain the way they solved one of the problems.

### Recommendations

- You can resubmit the assignment multiple times, so I would recommend submitting early and often. Better to have most of your assignment turned in in case of a technology failure than none at all.
- Due date and time is on a Sunday right before midnight. Although I try to do my best, the chances of getting any questions answered right before the due date and time can be small. Try to solve the homework with plenty of times to ask questions in the forum or, preferably, during class.

- You will always get a better credit for a wrong answer that is yours, than for a right answer that is someone else's. Make sure you add comments explaining your reasoning in code to get partial grade even when the problem does not return the correct result.
- Make sure that your output looks exactly like the output required. While we will have plenty of flexibility during the labs to experiment and explain the route we took, I may not be able to understand the reasoning behind changing the output in a homework, and that could result in lost points.
- Feel free to help each other, but do not copy from each other.
- Test your programs with many different inputs, make sure they return the correct result.

## Problems

1. (10 points) Write a program that uses a for statement to **calculate** (meaning, use C++ to do the calculation!) and print the product of the odd integers from 1 to 15. The screen dialog should appear as follows:

```
The product of odd integers from 1 to 15 is 2027025
```

2. (20 points) A mail order house sells five different products whose retail prices are: product 1 — \$2.98, product 2—\$4.50, product 3—\$9.98, product 4—\$4.49 and product 5— \$6.87. Write a program that reads a series of pairs of numbers as follows:

- a. product number
- b. Quantity sold

Your program should use a switch statement to determine the retail price for each product. Your program should calculate and display the total retail value of all products sold. Use a sentinel-controlled loop to determine when the program should stop looping and display the final results. The screen dialog should appear as follows:

```
Enter product number (or -1 to quit): 2
Enter quantity sold: 2
Enter product number (or -1 to quit): 3
Enter quantity sold: 10
Enter product number (or -1 to quit): -1
Total sales: 108.80
```

3. (30 points) A company pays its employees as managers (who receive a fixed weekly salary), hourly workers (who receive a fixed hourly wage for up to the first 40 hours they work and “time-and-a-half”—1.5 times their hourly wage—for overtime hours worked), commission workers (who receive \$250 plus 5.7 percent of their gross weekly sales), or pieceworkers (who receive a fixed amount of money per item for each of the items they produce—each pieceworker in this company works on only one type of item). Write a program to compute the weekly pay for each employee. You do not know the number of employees in advance. Each type of employee has its own pay code: Managers have code 1, hourly workers have code 2, commission workers have code 3 and pieceworkers

have code 4. Use a switch to compute each employee's pay according to that employee's paycode. Within the switch, prompt the user (i.e., the payroll clerk) to enter the appropriate facts your program needs to calculate each employee's pay according to that employee's paycode. The screen dialog should appear as follows:

```
Enter employee type (or -1 to quit): 1
Manager selected. Enter weekly salary: 1500
The manager's pay is: $1500
Enter employee type (or -1 to quit): 2
Hourly worker selected. Enter hourly salary: 15
Enter amount of hours worked: 42
The hourly worker's pay is: $645
Enter employee type (or -1 to quit): 3
Commission worker selected. Enter gross weekly sales: 100
The commission worker's pay is: $307
Enter employee type (or -1 to quit): 4
Pieceworker selected. Enter wage per piece: 10
Enter number of pieces: 25
The pieceworker's pay is: $250
Enter employee type (or -1 to quit): 5
Invalid worker code
Enter employee type (or -1 to quit): -1
```

4. (20 points) Function `floor` can be used to round a number to a specific decimal place. The statement

```
y = floor( x * 10 + .5 ) / 10,
```

rounds `x` to the tenths position (the first position to the right of the decimal point). The statement

```
y = floor( x * 100 + .5 ) / 100,
```

rounds `x` to the hundredths position (the second position to the right of the decimal point).

Write a program that defines four functions to round a number `x` in various ways:

- a. `roundToInteger( number )`
- b. `roundToTenths( number )`

- c. `roundToHundredths( number )`
- d. `roundToThousandths( number )`

For each value read, your program should print the original value, the number rounded to the nearest integer, the number rounded to the nearest tenth, the number rounded to the nearest hundredth and the number rounded to the nearest thousandth. The screen dialog should appear as follows:

```
Input a decimal number: 10.63678
Rounded to integer: 11
Rounded to tenths: 10.6
Rounded to hundredths: 10.64
Rounded to thousandths: 10.637
```

5. (20 points) Write a function that takes the time as three integer arguments (hours, minutes and seconds) and returns the number of seconds since the last time the clock “struck 12.” Use this function to calculate the amount of time in seconds between two times, both of which are within one 12-hour cycle of the clock. Note the first time can be later or earlier than the second time. The screen dialog should appear as follows:

```
Enter time one, hours: 9
Enter time one, minutes: 20
Enter time one, seconds: 30
Enter time two, hours: 8
Enter time two, minutes: 9
Enter time two, seconds: 10
Seconds in between: 4,280
```