CSCI 1300 - Intro to Computer Programming
Instructor: Fleming/Gupta
Recitation 3

This assignment is due **Saturday, February 3rd, by 6 pm**

- ***All components*** **(Cloud9 workspace, moodle quiz attempts, and zip file) must be completed and submitted by Saturday, February 3rd 6:00 pm for your solution to receive points.**
- **Recitation attendance is required to receive credit.**

Objectives:

- Understand basic programming concepts of: variables, values, assignment, operations, operands, and use them in simple C++ commands
- Understand and practice using `cout` to display text and variables (their values)
- Understand and practice using `cin` to take input from the user
- Write and test short C++ functions
  - write functions with zero, one, or more input parameters
  - write functions that have / do not have return values
- Understand the distinction between printing and returning values.

Please follow the same submission guidelines outlined in Homework 2 description regarding Style, Comments and Test Cases. Here's a review below on what you need to submit for Recitation 3.

Develop in Cloud9: For this recitation assignment, write and test your solution using Cloud9.
Submission: All three steps must be fully completed by the submission deadline for your homework to receive points. Partial submissions will not be graded.

1. ***Share your Cloud 9 workspace with your TA:*** Your recitation TA will review your code by going to your Cloud9 workspace. *TAs will check the last version that was saved before the submission deadline*.
   - Create a directory called **Rec3** and place all your file(s) for this assignment in this directory.
   - Share your workspace by clicking Share in the upper right hand corner and inviting your TA using their Cloud9 username.

TA Name	Cloud9 Username
Akriti Kapur	akritikapur
Arcadia(Xiaozhe) Zhang	arcadiaz
Ashwin Sankaralingam	ashwinsankaralingam
Bu Sun Kim	busun
Chelsea Chandler	chelseachandler
Harshini Muthukrishnan	harshini95

TA Name	Cloud9 Username
Monika Tak	mtak
Paramjot Singh	param17
Richard Tillquist	riti4538
Sayali Sonawane	sayalisonawane
Yichen Wang	wangyichen5151

- Make sure to *save* the final version of your code (File > Save). Verify that this version displays correctly by going to File > File Version History.
- The file(s) should have all of your functions, test cases for the functions in main function(s), and adhere to the style guide. Please read the **Test Cases** and **Style and Comments** sections for more details.

2. ***Submit to the Moodle Autograder:*** Head over to Moodle to the link **Rec 3**. You will find one programming quiz question for each problem in the assignment. Submit your solution for the first problem and press the Check button. You will see a report on how your solution passed the tests, and the resulting score for the first problem. You can modify your code and re-submit (press *Check* again) as many times as you need to, up until the assignment due date. Continue with the rest of the problems.
3. ***Submit a zip file to Moodle:*** After you have completed all the questions and checked them on Moodle, ***you must submit a zip file with the .cpp file(s) you wrote in Cloud9***. Submit this file going to **Rec 3 (File Submission)** on moodle.

Problems Set:
Write a **function** for each of the following problems. You should first write your solution in Cloud9. Then copy and paste your function into the moodle quiz questions.

Note: You are required to also submit a .cpp file with your functions and a `main()` function with 2 tests for each of your functions. Because of the related nature of these 3 problems, you are encouraged to use output from one function as input into another when writing your tests in `main()`.

Problem 1
Write a function to prompt the user for the number of hours they have worked this week using `cin`.

- Your function should take **no** input parameters/arguments
- Your function should **return** the number of hours the user has worked. **Fractional values are allowed.**
- Your function should use the following prompt for user input: "**Please enter how many hours you worked this week: "**
- Your function *MUST* be named **getHours**.

For example, if a user enters 36.5 for hours worked, your function should return 36.5

CSCI 1300 - Intro to Computer Programming
Instructor: Fleming/Gupta
Recitation 3

Problem 2

Your function should take in two floating point parameters representing the number of hours worked and an hourly pay rate and return the calculated total pay.

- Your function should have **two** input values:
    - a floating point parameter for hours worked
    - a floating point parameter for pay rate
- Your function should not print/display/output anything.
- Your function should **return** the calculated pay.
- Your function *MUST* be named **calcPay**.

Problem 3

Write a function to output (print) the user's pay.

Expected Output:

```
Your pay is $200.
```

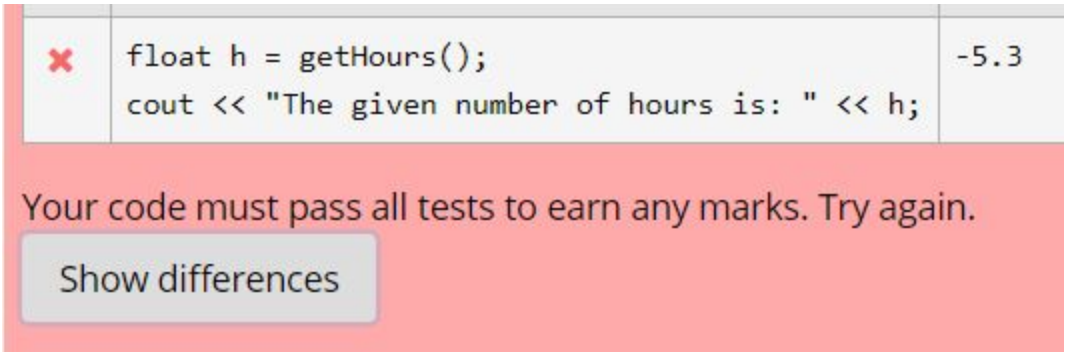
- Your function should take in one input parameter/argument, a float value for the pay.
- Your function should not **return** anything.
- Your function *MUST* be named **printPay**.

Submitting Your Code to the Autograder on Moodle

You must name the functions as indicated in each problem description. **Importantly**, *the cout formats provided for each problem are not suggestions – they **MUST** be followed precisely, word-for-word and including all punctuation marks*, otherwise the autograder will not recognize your results and you will not receive credit.

If there are errors in your solution to a particular problem, a button labeled “Show differences” will appear below the table of tests after you hit “check”. This can be a very useful tool in helping you find small typos, especially in cout statements.

CSCI 1300 - Intro to Computer Programming
Instructor: Fleming/Gupta
Recitation 3



✖	`float h = getHours();` `cout << "The given number of hours is: " << h;`	-5.3

Your code must pass all tests to earn any marks. Try again.

Show differences

For example, below we hit "check" for a solution to problem 1 on this recitation and have failed all the test cases despite getting the correct values. Hitting "Show differences", we can see that a colon (:) is missing at the end of our prompt. When characters are in the expected output but not in your output they are highlighted in the "Expected" column.

Expected	Got	
Please enter how many hours you worked this week:↩ The given number of hours is: 36.5	Please enter how many hours you worked this week↩ The given number of hours is: 36.5	✖
Please enter how many hours you worked this week:↩ The given number of hours is: 12	Please enter how many hours you worked this week↩ The given number of hours is: 12	✖
Please enter how many hours you worked this week:↩ The given number of hours is: 0	Please enter how many hours you worked this week↩ The given number of hours is: 0	✖
Please enter how many hours you worked this week:↩ The given number of hours is: -5.3	Please enter how many hours you worked this week↩ The given number of hours is: -5.3	✖

On the other hand, when we include extra, unexpected characters in output they are highlighted in the "Got" column. Below we added the word "have" plus a space to the prompt.

Expected	Got	
Please enter how many hours you worked this week:↩ The given number of hours is: 36.5	Please enter how many hours you have worked this week:↩ The given number of hours is: 36.5	✖
Please enter how many hours you worked this week:↩ The given number of hours is: 12	Please enter how many hours you have worked this week:↩ The given number of hours is: 12	✖
Please enter how many hours you worked this week:↩ The given number of hours is: 0	Please enter how many hours you have worked this week:↩ The given number of hours is: 0	✖
Please enter how many hours you worked this week:↩ The given number of hours is: -5.3	Please enter how many hours you have worked this week:↩ The given number of hours is: -5.3	✖

You are required to also submit a .cpp file with your functions and a `main()` function with 2 tests for each of your functions.