

CSCI 1300 CS1: Starting Computing

Ashraf, Cox, Spring 2020

Homework 3

Due: Saturday, February 8, by 6 pm

(5 % bonus on the total score if submitted by 11:59 pm February 7)

Objectives

- Understand how to implement decisions using if statements.

You can find [hw3 note: if-else, switch on Moodle](#).

Submissions

- [Conceptual reviews\(mcq\)](#). There are a few multiple-choice questions to check your conceptual understanding. Don't forget to complete them!
- [C++ files](#). All files should be named as specified in each question, and they should compile and run on Cloud 9 to earn full points. TAs will be grading styles of your code and comments. Please see [the style guide on Moodle](#) and [the summary note on Moodle](#). At the top of each file, write your name with the following format:

```
// CS1300 Spring 2020
// Author: Punith Sandhu
// Recitation: 123 - Favorite TA
// Homework 3 - Problem # ...
```

- [Code runner](#). Your program will be graded by Code Runner. You can modify your code and re-submit (press "Check" again) as many times as you need to, until the assignment due date.

Plan for this week

Questions 1 - 3 are considered recitation questions. You should be able to finish at least 3 questions during the recitation time. If you're able to finish them earlier, then continue working on the other questions and complete the homework before the deadline.

Questions

Question 1(5pt): Practicum average

In this course, students need to have at least 67 average score on practicums to earn C- or higher letter grades. It would be useful to have a program to calculate your practicum average. Write a program that asks the user to enter three practicum scores and calculate the average score. Then, it prints the average score and prints the message depending on the average score.

If the calculated average score is 67 or higher, then it prints

`"You have a passing practicum average."`

If it's below 67, then it prints

`"You can retake practicums during the final."`

Each practicum score should be an integer value. The practicum average should be a double formatted with a two-digit precision as shown below.

Expected output 1 (**bold** is user input)

```
Enter practicum 1 score:
90
Enter practicum 2 score:
85
Enter practicum 3 score:
75
Practicum average: 83.33
You have a passing practicum average.
```

The file should be named as `practicum.cpp`. Don't forget to head over to the code runner on Moodle and paste your solution in the answer box!

Question 2(10pt): Total pay

Write a program that asks the number of hours worked and hourly pay rate, as floating point numbers. Then, it displays the calculated total pay. Hours worked over 40 hours are considered as overtime and should be calculated at 1.5x the pay rate.

Be sure to have input validation. If one or both input values are negative values, it does not make sense. If a pay rate of -\$10 and 41 hours worked, then the pay would be -\$415. Instead of printing the negative value of the pay, print `"Invalid input"`.

For example, if Punith worked 42 hours (40 hours + 2 hours of overtime), and his pay rate is \$12. Then, his paycheck will be \$516.

Regular Pay:	\$ 480.00	(\$12 x 40 hours)
Overtime Pay:	\$ 36.00	(1.5 x \$12 * 2 hours)
Total Pay:	\$ 516.00	

The total pay(pay check) should be formatted with a two-digit precision as shown below.

Expected output 1 (**bold** is user input)

```
Enter the number of hours worked
45.5
Enter an hourly pay rate
17.6
Pay check: 849.20
```

Expected output 2 (**bold** is user input)

```
Enter the number of hours worked
-45.5
Enter an hourly pay rate
20.0
Invalid input
```

The file should be named as `pay.cpp`. Don't forget to head over to the code runner on Moodle and paste your solution in the answer box!

Question 3(10pt): Sushi restaurant

Ashwin opened a sushi restaurant in Seattle. Each sushi piece is \$1.99. His restaurant offers a student discount depending on the number of pieces they eat.

Quantity	Discount
10 - 19	10 %
20 - 49	15 %
50 - 99	20 %
100 or more	25 %

Write a program that asks the number of sushi pieces eaten and compute the total price. Be sure to have an input validation. The value entered should be greater than 0. If not, print "Invalid input". The total price should be formatted with a two-digit precision as shown below.

Expected output 1 (**bold** is user input)

```
Enter the number of sushi:
3
Total price: $5.97
```

Expected output 2 (**bold** is user input)

```
Enter the number of sushi:
-7
Invalid input
```

Expected output 3 (**bold** is user input)

```
Enter the number of sushi:  
23  
Total price: $38.90
```

The file should be named as `sushi.cpp`. Don't forget to head over to the code runner on Moodle and paste your solution in the answer box!

Question 4(10pt): boxers' weights

Your manager gave a task to write a program that takes weights of three boxers (as integer values) and check if they are sorted by their weights. If the weights are sorted, then it prints "Ascending order", "Descending order", or "Neither". If two or more weights are the same, they are considered as neither. The users are well trained, and they will enter weights separated by spaces.

Expected output 1 (**bold** is user input)

```
Enter boxers' weights  
162 180 225  
Ascending order
```

Expected output 2 (**bold** is user input)

```
Enter boxers' weights  
225 180 162  
Descending order
```

Expected output 3 (**bold** is user input)

```
Enter boxers' weights  
162 135 225  
Neither
```

The file should be named as `boxer.cpp`. Don't forget to head over to the code runner on Moodle and paste your solution in the answer box!

Question 5(10pt): Days in a month

Write a program that asks the user to enter a year and month(1-12). Then, the program should display the number of days in that month. If the entered month is not between 1 and 12, then print "Invalid month". Don't forget about leap years!

What is a leap year? In general, years divisible by 4 are leap years. For dates after 1582, however, there is a Gregorian correction: years that are divisible by 100 are not leap years but years divisible by 400, are. So, for instance, 1900 was not a leap year but 2000 was.

Expected output (**bold** is user input)

```
Enter a year:
2008
Enter a month:
2
29 days
```

Extra credit: if you use a `switch` statement for this question, we will give you 5pt extra credit.

The file should be named as `daysInMonth.cpp`. Don't forget to head over to the code runner on Moodle and paste your solution in the answer box!

Question 6(15pt): Personalized ad

You have graduated from CU Boulder, and now have a job to create a personalized ad program to serve a person on a social media platform. The program prompts the user for information and then text that describes ads based on their inputs. There are three scenarios for the program. The program asks the users which scenario they want to do. Then, depending on the choices, the program asks information and show an ad. Please make sure to print `"Invalid option."` if they choose other than three scenarios.

Extra Credit: If you use a `switch` statement for this question, we will give you 5 pt extra credit.

Scenario 1: For advertisers using the lowest tier of personalization on this social media platform, there is limited data about users available — we only know whether a user self-reported that they own dog.

For scenario 1, the program asks the user to enter if they own a dog. If they own a dog, then the program prints:

```
"Meat's the need of dogs like yours!"
```

Otherwise, the program prints:

```
"Dirty mouth? Clean it up with new Orbit Raspberry Mint."
```

The users will be more likely to type `"yes"` or `"no"`, but some of them might not. We consider `"yes"` and `"Yes"` as `"yes"`. If the users type other than `"yes"` or `"Yes"`, then the program considers them as a `"no"`.

Expected output for Scenario 1 (**bold** is user input):

```
How personalized should the ad be? (1-3)
1
Do you own a dog?
yes
Meat's the need of dogs like yours!
```

Scenario 2: Combining data from a marketing platform with the results from personality tests, we know that people with at least 500 Facebook friends are likely to be extroverts, and people with less than 500 friends are likely to be introverts. By using computer-vision, we also know extroverts post more dog photos, and introverts tend to post more photos of cats. Market research has shown that dog people are highly likely to be positively influenced by advertisements for any product that includes dogs, and similar for cat people and cats.

For Scenario 2, the program asks the user how many Facebook friends they have. If the number of Facebook friends is greater than or equal to 500, the program prints:

```
"Don't like cleaning up after your dog? Call Doody Free to sign up
for backyard cleanup or dog walking service."
```

If the number of Facebook friends is less than 500, it prints:

```
"Who doesn't need another cat? Adopt a shelter pet right meow."
```

Expected output for Scenario 2 (**bold** is user input):

```
How personalized should the ad be? (1-3)
2
How many Facebook friends do you have?
450
Who doesn't need another cat? Adopt a shelter pet right meow.
```

Scenario 3: Based on ad clicks and third-party data shared through purchases on a "marketplace" platform, we have data about likely income averages for every zip code in the country. We also know based on self-reported age how incomes vary based on life stage (e.g., college students have less disposable income no matter where they live). Advertisers strategically advertise to people who are more likely to be able to afford their product.

For Scenario 3, the program asks the users to enter their zip code and age. Then, it prints the ad.

- If the zipcode is in Boulder (i.e. between and including 80301 and 80310) and the user is at least 25, print ad text for an expensive product.

```
"You are surrounded by professional athletes. Up your exercise
game in the privacy of your home with a Peloton bike."
```

- If the zipcode is in Boulder and the user is younger than 25, print ad text for an inexpensive product.
 “Looking for dinner that won’t break the bank? Tacos. Come grab a \$3 at Centro Latin Kitchen.”
- Otherwise, print ad text encouraging the user to visit Boulder.
 “Feeling Uninspired? The Boulder Flatirons are calling. Come hike to inspiring views, delicious food, and enjoy the hospitality of Boulder, CO.”

Expected output for Scenario 3 (**bold** is user input):

```
How personalized should the ad be? (1-3)
3
What is your zip code?
80302
What is your age?
22
Looking for dinner that won’t break the bank? Tacos. Come grab a $3
at Centro Latin Kitchen.
```

Invalid Scenario: If the user chooses a scenario that is not between 1 and 3, print “Invalid option.”

Expected output for an invalid scenario (**bold** is user input):

```
How personalized should the ad be? (1-3)
7
Invalid option.
```

The file should be named as `personalizedAd.cpp`. Don’t forget to head over to the code runner on Moodle and paste your solution in the answer box!

Homework 3 checklist

Here is a checklist for submitting the assignment:

1. Complete the [conceptual reviews\(mcq\)](#)
2. Complete the code [Homework 3 CodeRunner](#)
3. Submit one zip file to [Homework 3](#). The zip file should be named, **hmkw3_lastname.zip**. It should have the following 6 files:
 - **practicum.cpp**
 - **pay.cpp**
 - **sushi.cpp**
 - **boxer.cpp**
 - **daysInMonth.cpp**
 - **personalizedAd.cpp**

Homework 3 points summary

Criteria	Pts
Conceptual reviews (MCQ)	10
CodeRunner (problem 1 - 6)	60
C++ file submission (compiles and runs, style and comments)	30
<hr/>	
Recitation attendance (Week 4)*	-30
Total	100
5% early submission bonus	+5%
Implement with <code>switch</code> (problem 5, 6)	+5pt each

* if your attendance is not recorded, you will lose points. Make sure your attendance is recorded on Moodle.