Q1

1 Point

What is the main difference between classes and structs in C++?

○ Only classes can have pointers as member variables

● Classes by default have private members while structs by default have public members

○ There can be pointers to classes but not to structs

○ Structs do not have destructors

Q2

4 Points

What are the consequences of having a poorly detailed commit message? How would this affect a team of developers working together on the same repository?

The consequences of having a poorly detailed commit message would affect the teammates' understanding of the commit and therefore cause some problem when error happened and they need to rollback.

Q3

10 Points

In C++, write (that is, declare and define) a class named *personWeight* that stores the data necessary to represent the weight of a person in pounds and ounces.

The *personWeight* class has private member variables named *pounds* (which can store only whole numbers, such as 5), and *ounces* (which can store either whole numbers such as 5 or decimal numbers such as 1.5)

The *personWeight* class has a public function (method) named *weight_in_ounces* that returns the weight of a person in ounces. The formula for computing the weight of a person in ounces is:

(16 * pounds) + ounces

NOTE: the declaration of the *weight_in_ounces* method should reside inside the declaration/definition of the *personWeight* class, and the definition of the *weight_in_ounces* function (method) should reside outside of the declaration/definition of the *personWeight* class.

NOTE: Do not write any constructors for the class here. You will do that in the next question.

Be careful of the variable types declarations you use to declare pounds and ounces (and the type of value returned by the weight_in_ounces method)

```
class personWeight {
  public:
    float weight_in_ounces();

  private:
    int pounds;
    float ounces;
}

float weight_in_ounces(int pounds, float ounces) {
   return (float) 16 * this->pounds + this->ounces;
}
```

Q4

5 Points

In C++, write (declare and define) ONE and only ONE constructor for the *personWeight* class that resides inside of the *personWeight* class definition/declaration and can be called as:

A parameterless constructor (i.e., with no parameters), to instantiate a personWeight object.

or

A constructor which accepts the following two parameters named pounds and ounces, which store whole numbers (e.g., 180) and decimal numbers (e.g., 8.75) respectively.

The default values of the pounds and ounces parameters should be set to zero. The constructor should initialize the private variables pounds and ounces to either the values passed in via the parameters pounds and ounces, or zero (0) if no value is passed in via the parameters pounds and/or ounces.

You may assume your answer will be included in the class declaration from question 2 above. Do not rewrite the class declaration/definition from the above question.

```
class personWeight {
   public:
      personWeight();
      personWeight(int pounds, float ounces);
}

personWeight::personWeight() { // default constructor
   pounds = 0;
   ounces = 0.0;
}

personWeight::personWeight(int pounds, float ounces) {
   this->pounds = pounds;
   this->ounces = ounces;
}
```

Q5

5 Points

Next write a C++ main function that does the following:

1. Creates an instance of the *personWeight* class named Tim with a weight set to 203 pounds and 12.25 ounces.
2. Creates an instance of the *personWeight* class named Dan with a weight set to 180 pounds and 11.25 ounces
3. Creates an instance of the *personWeight* class named Lightweight with the default values (so no parameters are passed to the constructor).

4. Prints out Dan’s weight in ounces.
5. Prints out Tim’s weight in ounces.
6. Prints out Lightweight’s weight in ounces.

Note: Getters/Setters do not need to be written to solve this question.

You can assume the declaration/definition of the *personWeight* class, and the definition of the *weight_in_ounces* function has been correctly defined in the same file as the main program you are to write, and that the macro *#include* is the first line at the top of the file.

Statements and objects that you may find useful:

using namespace std;

The object *cout* and the object *endl* from the std library

```
using namespace std;
#include <string>

personWight Tim = new Tim(203, 12.25);
personWight Dan = new Tim(180, 11.25);
personWight LightWeight = new Tim();
cout << Dan->weight_in_ounces() << endl;
cout << Tim->weight_in_ounces() << endl;
cout << LightWeight->weight_in_ounces() << endl;
```

Quiz 01

● GRADED

STUDENT
Ruijun Ni

TOTAL POINTS
17 / 25 pts

QUESTION 1	(no title)	**1** / 1 pt
QUESTION 2	(no title)	**4** / 4 pts
QUESTION 3	(no title)	**8** / 10 pts
QUESTION 4	(no title)	**0** / 5 pts
QUESTION 5	(no title)	**4** / 5 pts