

## Solutions of C++ Questions

### 1. Question 1

a) work.h

```
/**
 * @file work.h
 * @author tamersum11
 * @brief Class Work whose objects represent working times (in whole minutes) and
 * salary rates (in whole cents per minute).
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 */
#ifndef WORK_H
#define WORK_H

#include <iostream>

class Work
{
public:
    Work(int salaryRate = 0, int workingTimes = 0);
    Work(const Work *work);
    ~Work();
    void add(int workingTimes = 0);
    void printSalary(void);
    bool subtract(int workingTimes = 0);
    int compare(const Work *work);
    static void reset(Work *work = nullptr);

private:
    int salaryRate, workingTimes;
};

#endif // WORK_H
```

b) work.cpp

```
/**
 * @file work.cpp
 * @author tamersum11
 * @see Work.h
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 */
#include "work.h"

/**
 * @brief Construct a new Work:: Work object
 *
 * @param salaryRate in whole cents per minute
 * @param workingTimes represent working times (in whole minutes)
 */
Work::Work(int salaryRate, int workingTimes)
{
    this->salaryRate = salaryRate;
    this->workingTimes = workingTimes;
}

/**
 * @brief Construct a new copy of Work:: Work object
 *
 * @param work
 */
Work::Work(const Work *work)
{
    this->salaryRate = work->salaryRate;
    this->workingTimes = work->workingTimes;
}
```

```

/**
 * @brief Destroy the Work:: Work object
 *
 */
Work::~Work(){}

/**
 * @brief Adds the given minutes to working time
 *
 * @param workingTimes time to be added
 */
void Work::add(int workingTimes)
{
    this->workingTimes += workingTimes;
}

/**
 * @brief Prints salary in dollars. (salaryRate * workingTimes) / 100
 *
 */
void Work::printSalary()
{
    std::cout << (float)(this->workingTimes * this->salaryRate) / 100 << std::endl;
}

/**
 * @brief Attempts to subtract the given minutes to working time
 *
 * @param workingTimes time to be subtracted
 * @return true if there is sufficient time available (time remains changed)
 * @return false if not sufficient time available (time remains unchanged)
 */
bool Work::subtract(int workingTimes)
{
    if(this->workingTimes - workingTimes < 0) {
        return false;
    } else {
        this->workingTimes -= workingTimes;
        return true;
    }
}

```

```

/**
 * @brief Compares salaries the given work object to current work object
 *
 * @param work object pointer to be compared
 * @return 0 if salaries of object and compared object are equal
 * @return 1 if objects salary is bigger
 * @return -1 else
 */
int Work::compare(const Work *work)
{
    float salaryThis = (float)(this->workingTimes * this->salaryRate) / 100;
    float salaryWork = (float)(work->salaryRate * work->workingTimes) / 100;

    if(salaryThis == salaryWork) {return 0;}
    else if(salaryThis > salaryWork) {return 1;}
    else {return -1;}
}

/**
 * @brief Reset working time to zero
 *
 * @param work object pointer to be reset
 */
void Work::reset(Work *work)
{
    work->workingTimes = 0;
}

```

c) main.cpp

```

/**
 * @file main.cpp
 * @author tamersum11
 * @brief Tests the class Work public member functions
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 */

```

Address: Istanbul, TURKEY

GSM: +90 533 347 6148 | <https://github.com/tamersum11>

```

#include <iostream>
#include "work.h"

using namespace std;

int main()
{
    Work *w = new Work(25, 60);
    w->add(65);
    w->printSalary();
    Work::reset(w);
    bool okay = w->subtract(60);
    cout << okay << endl;

    Work *v = new Work(30);
    int r = w->compare(v);
    cout << r << endl;

    Work u(v);
    u.printSalary();

    delete w;
    w = nullptr;
    delete v;
    v = nullptr;

    return 0;
}

```

d) Output:

```

Starting C:\Users\Tamer\Desktop\Foreks-
C++Questions\Solutions\Question1\build-question1-
Desktop_Qt_6_2_1_MinGW_64_bit-Debug\question1.exe ...
31.25
0
0
0

```

## 2. Question 2

a) Output:

Calls Base() for object a	-> Base
Calls Base() for object d	-> Base
Calls Der(int i) for object d	-> Der2
Calls ~Der() for object d	-> Destruct Der
Calls ~Base() for object d	-> Destruct Base
Calls ~Base() for object a	-> Destruct Base

### 3. Question 3

a) quad.h

```
/**
 * @file quad.h
 * @author tamersum11
 * @brief Abstract base class Quad have four member data variables (floats)
representing side lengths
 * and a pure virtual function Area. It also have a method for setting the data
variables.
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#ifndef QUAD_H
#define QUAD_H

class Quad
{
public:
    virtual float Area() = 0;
    void setSides(float side_1, float side_2, float side_3, float side_4) {
        this->side_1 = side_1;
        this->side_2 = side_2;
        this->side_3 = side_3;
        this->side_4 = side_4;
    }
protected:
    float side_1, side_2, side_3, side_4;
};

#endif // QUAD_H
```

b) rectangle.h

```
/**
 * @file rectangle.h
 * @author tamersum11
 * @brief Derive a class Rectangle from Quad and override the Area method so that it
returns the area of the Rectangle.
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#ifndef RECTANGLE_H
#define RECTANGLE_H

#include "quad.h"

class Rectangle : public Quad
{
public:
    float Area() override;
};

#endif // RECTANGLE_H
```



### c) rectangle.cpp

```
/**
 * @file rectangle.cpp
 * @author tamersum11
 * @see rectangle.h
 * @see quad.h
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#include "rectangle.h"

/**
 * @brief Overrides the Area method from base class Quad
 *
 * @return float the area of the Rectangle
 */
float Rectangle::Area()
{
    return side_1 * side_2;
}
```

d) main.cpp

```
/**
 * @file main.cpp
 * @author tamersum11
 * @brief
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#include <iostream>
#include "rectangle.h"

using namespace std;

/**
 * @brief Top-level function that will take a parameter of type Quad and return the
value of the appropriate Area function.
 *
 * @param quad
 * @return float the value of the appropriate Area function
 */
float getArea(Quad *quad) {
    return quad->Area();
}

int main()
{
    Quad *rectangle = new Rectangle();
    rectangle->setSides(3.00, 4.00, 3.00, 4.00);

    cout << getArea(rectangle) << endl;

    return 0;
}
```

e) Output:

```
Starting C:\Users\Tamer\Desktop\Foreks-  
C++Questions\Solutions\Question3\build-question3-  
Desktop_Qt_6_2_1_MinGW_64_bit-Debug\question3.exe ...  
12
```

#### 4. Question 4

- a) Compile time error: There cannot be instanced to an abstract base class
- b) There is no error
- c) There is no error
- d) There is no error
- e) There is no error
- f) There is no error
- g) Compile time error: There cannot initialize I\* type with E\* type.
- h) There is no error
- i) Compile time error: There cannot initialize D\* type with its base C\* type.
- j) Runtime error: Downcasting is invalid type cast for from derived class D to its base class C.
- k) There is no error
- l) There is no error
- m) There is no error
- n) There is no error
- o) Compile time error: There cannot be a conversion from E\* to I\*.

## 5. Question 5

a) triple.h

```
/**
 * @file triple.h
 * @author tamersum11
 * @brief Class named Triple with three private data members (floats) x, y, and z.
 * It provides public functions for setting and getting values of all the private
data members.
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#ifndef TRIPLE_H
#define TRIPLE_H

#include <iostream>

class Triple
{
public:
    Triple(float x = 0.00, float y = 0.00, float z = 0.00);
    ~Triple();

    //Setters and getters
    float getX() const;
    void setX(float newX);
    float getY() const;
    void setY(float newY);
    float getZ() const;
    void setZ(float newZ);

    //Overloaded operators:
    Triple& operator=(const Triple &obj);
    Triple& operator++();
    Triple operator++(int);
    Triple& operator()(float x, float y, float z);
};
```

Address: Istanbul, TURKEY

---

GSM: +90 533 347 6148 | <https://github.com/tamersum11>

```

/**
 * @brief Addition so that corresponding elements are added together
 *
 * @param obj1
 * @param obj2
 * @return Triple
 */
friend Triple operator+(const Triple &obj1, const Triple &obj2) {
    return Triple(obj1.x + obj2.x, obj1.y + obj2.y, obj1.z + obj2.z);
}

/**
 * @brief Output so that it displays the Triple in the form "The triple is (x,
y, z)."
 *
 * @param os
 * @param obj
 * @return std::ostream&
 */
friend std::ostream& operator<<(std::ostream& os, const Triple &obj) {
    os << "The triple is (" << obj.x
        << "," << obj.y << "," << obj.z << ").";
    return os;
}

private:
    float x, y, z;
};

/**
 * @brief Assignment that copies x to z, y to x, and z to y
 *
 * @param obj
 * @return Triple&
 */
inline Triple &Triple::operator=(const Triple &obj)
{
    z = obj.getX();
    x = obj.getY();
    y = obj.getZ();

    return *this;
}

```

```

/**
 * @brief Pre-increment so that x and z are increased by one each
 *
 * @return Triple&
 */
inline Triple &Triple::operator++() {
    this->x++;
    this->z++;
    return *this;
}

/**
 * @brief Post-increment so that x and z are increased by one each.
 *
 * @return Triple
 */
inline Triple Triple::operator++(int) {
    Triple old = *this;
    operator++();
    return old;
}

/**
 * @brief Function call operator so that the values for x, y and z can be set
 *
 * @param x
 * @param y
 * @param z
 * @return Triple&
 */
inline Triple &Triple::operator()(float x, float y, float z)
{
    setX(x);
    setY(y);
    setZ(z);

    return *this;
}

#endif // TRIPLE_H

```

b) triple.cpp

```
/**
 * @file triple.cpp
 * @author tamersum11
 * @see triple.h
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#include "triple.h"

/**
 * @brief Construct a new Triple:: Triple object or, by default, sets the values all
equal to 0
 *
 * @param x
 * @param y
 * @param z
 */
Triple::Triple(float x, float y, float z)
{
    this->x = x;
    this->y = y;
    this->z = z;
}

/**
 * @brief Destroy the Triple:: Triple object
 *
 */
Triple::~Triple(){}

```

```

/**
 * @brief Getter for x
 *
 * @return float x
 */
float Triple::getX() const
{
    return x;
}

/**
 * @brief Setter for x
 *
 * @param newX
 */
void Triple::setX(float newX)
{
    x = newX;
}

/**
 * @brief Getter for y
 *
 * @return float y
 */
float Triple::getY() const
{
    return y;
}

/**
 * @brief Setter for y
 *
 * @param newY
 */
void Triple::setY(float newY)
{
    y = newY;
}

```



```

/**
 * @brief Getter for z
 *
 * @return float z
 */
float Triple::getZ() const
{
    return z;
}

/**
 * @brief Setter for z
 *
 * @param newZ
 */
void Triple::setZ(float newZ)
{
    z = newZ;
}

```

c) main.cpp

```

/**
 * @file main.cpp
 * @author tamersum11
 * @brief Tests the overloaded operators of Triple class
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#include <iostream>
#include "triple.h"

using namespace std;

```

Address: Istanbul, TURKEY

GSM: +90 533 347 6148 | <https://github.com/tamersum11>

```

int main()
{
    cout << "Triple t(1.00):" << endl;
    Triple t(1.00);
    cout << t << endl << endl;

    cout << "t(1.00, 2.00, 3.00):" << endl;
    t(1.00, 2.00, 3.00);
    cout << t << endl << endl;

    cout << "Triple t1(1.00, 2.00, 3.00):" << endl;
    Triple t1(1.00, 2.00, 3.00);
    cout << t1 << endl << endl;

    cout << "Triple t2(1.00, 2.00, 3.00):" << endl;
    Triple t2(1.00, 2.00, 3.00);
    cout << t2 << endl << endl;

    cout << "Triple t3(1.00, 2.00, 3.00):" << endl;
    Triple t3(1.00, 2.00, 3.00);
    cout << t3 << endl << endl;

    cout << "t1 + t2 + t3:" << endl;
    cout << t1 + t2 + t3 << endl << endl;

    cout << "t = t1 + t2 + t3:" << endl;
    t = t1 + t2 + t3;
    cout << t << endl << endl;

    cout << "++t:" << endl;
    ++t;
    cout << t << endl << endl;

    cout << "t++:" << endl;
    t++;
    cout << t << endl;

    return 0;
}

```

d) Output:

Starting C:\Users\Tamer\Desktop\Foreks-  
C++Questions\Solutions\Question5\build-question5-  
Desktop\_Qt\_6\_2\_1\_MinGW\_64\_bit-Debug\question5.exe ...

Triple t(1.00):

The triple is (1,0,0).

t(1.00, 2.00, 3.00):

The triple is (1,2,3).

Triple t1(1.00, 2.00, 3.00):

The triple is (1,2,3).

Triple t2(1.00, 2.00, 3.00):

The triple is (1,2,3).

Triple t3(1.00, 2.00, 3.00):

The triple is (1,2,3).

t1 + t2 + t3:

The triple is (3,6,9).

t = t1 + t2 + t3:

The triple is (6,9,3).

++t:

The triple is (7,9,4).

t++:

The triple is (8,9,5).

## 6. Question 6

a) The lines that will not compile in the main function:

- > A objA;
- > B objB;
- > C.setX(2);
- > cout<<C.getX();
- > D.setX(1);
- > D.f(3);

## 7. Question 7

a) Output:

For (pa=&a), output line of (pa -> f())	-> 1
For (pa=&a), output line of (pa -> g())	-> 2
For (pa=&b), output line of [(pa -> f())(1) + (pa -> g())(4)]	-> 5
For (pa=&c), output line of (pa -> f())	-> 1
For (pa=&c), output line of (pa -> g())	-> 5

## 8. Question 8

a) point.h

```
/**
 * @file point.hpp
 * @author tamersum11
 *
 * @brief Template class Point with two class parameters representing the two
coordinates of the Point.
 * Include public methods to display and set the data values as well as a function
that swaps the values so that,
 * after the swap, the first element is cast into the second and the second is cast
into the first.
 *
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 */
#ifndef POINT_H
#define POINT_H

#include <iostream>

template <typename T>
class Point
{
private:
    T x, y;

public:
    Point(T x, T y);
    Point();
    ~Point();

    //Setters and Getters
    void setX(T newX);
    T getX(void) const;
    void setY(T newY);
    T getY(void) const;
};
```

Address: Istanbul, TURKEY

---

GSM: +90 533 347 6148 | <https://github.com/tamersum11>

```
    void display(void);  
    void swap(void);  
};  
  
#endif // POINT_H
```

b) point.cpp

```
/**  
 * @file point.cpp  
 * @author tamersum11  
 * @see point.h  
 * @version 0.1  
 * @date 2022-02-13  
 *  
 * @copyright Copyright (c) 2022  
 */  
#include "point.h"  
  
/**  
 * @brief Construct a new Point< T>:: Point object  
 *  
 * @tparam T  
 * @param x  
 * @param y  
 */  
template<typename T>  
Point<T>::Point(T x, T y)  
{  
    this->x = x;  
    this->y = y;  
}  
  
/**  
 * @brief Construct a new Point< T>:: Point object  
 *  
 * @tparam T  
 */  
template<typename T>  
Point<T>::Point(){}  
}
```

Address: Istanbul, TURKEY

---

GSM: +90 533 347 6148 | <https://github.com/tamersum11>

```

/**
 * @brief Destroy the Point< T>:: Point object
 *
 * @tparam T
 */
template<typename T>
Point<T>::~~Point(){}

/**
 * @brief Sets the value of x coordinate of the Point
 *
 * @tparam T
 * @param newX
 */
template<typename T>
void Point<T>::setX(T newX)
{
    x = newX;
}

/**
 * @brief Gets the value of x coordinate of the Point
 *
 * @tparam T
 * @return T
 */
template<typename T>
T Point<T>::getX() const
{
    return x;
}

/**
 * @brief Sets the value of y coordinate of the Point
 *
 * @tparam T
 * @param newY
 */
template<typename T>
void Point<T>::setY(T newY)
{
    y = newY;
}

```

```

/**
 * @brief Gets the value of x coordinate of the Point
 *
 * @tparam T
 * @return T
 */
template<typename T>
T Point<T>::getY() const
{
    return y;
}

/**
 * @brief Displays the data values of two coordinates of the Point
 *
 * @tparam T
 */
template<typename T>
void Point<T>::display()
{
    std::cout << "x: " << x << std::endl
                << "y: " << y << std::endl;
}

/**
 * @brief Swaps the values so the first element is cast into the second
 * and the second is cast into the first
 *
 * @tparam T
 */
template<typename T>
void Point<T>::swap()
{
    std::swap(x, y);
}

/**
 * @brief The explicit instantiation of template class named Point
 */
template class Point<int>;
template class Point<float>;
template class Point<double>;
template class Point<long long>;

```



c) main.cpp

```
/**
 * @file main.cpp
 * @author tamersum11
 * @brief Tests the template class named Point
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#include <iostream>
#include "point.h"

using namespace std;

int main()
{
    int x = 5, y = 4;
    float x1 = 5.324, y1 = 4.134;

    Point<int> *a = new Point<int>(x, y);
    a->display();
    a->swap();
    a->display();

    Point<float> *b = new Point<float>();
    b->setX(x1);
    b->setY(y1);
    b->display();
    b->swap();
    b->display();

    delete a;
    a = nullptr;
    delete b;
    b = nullptr;

    return 0;
}
```

d) Output:

```
Starting C:\Users\Tamer\Desktop\Foreks-  
C++Questions\Solutions\Question8\build-question8-  
Desktop_Qt_6_2_1_MinGW_64_bit-Debug\question8.exe ...  
x: 5  
y: 4  
x: 4  
y: 5  
x: 5.324  
y: 4.134  
x: 4.134  
y: 5.324
```

## 9. Question 9

a) existence.h

```
/**
 * @file existence.hpp
 * @author tamersum11
 *
 * @brief Class Existence that contains two class data members numBorn and
numLiving.
 * The value of numBorn will be equal to the number of objects of the class that
have been instantiated.
 * The value of numLiving will be equal to the total number of objects in existence
currently.
 * (ie, the objects that have been constructed but not yet destructed.)
 *
 *
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#ifndef EXISTENCE_H
#define EXISTENCE_H

class Existence
{
private:
    static unsigned int numBorn;
    static unsigned int numLiving;
public:
    Existence();
    ~Existence();

    unsigned int getNumBorn(void) const;
    unsigned int getNumLiving(void) const;
};

#endif // EXISTENCE_H
```

Address: Istanbul, TURKEY

GSM: +90 533 347 6148 | <https://github.com/tamersum11>

b) existence.cpp

```
/**
 * @file existence.cpp
 * @author tamersum11
 * @see existence.hpp
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#include "existence.h"

/**
 * @brief Initializing the static variables
 *
 */
unsigned int Existence::numBorn = 0;
unsigned int Existence::numLiving = 0;

/**
 * @brief Construct a new Existence:: Existence object
 *
 */
Existence::Existence() {
    numBorn++;
    numLiving++;
}

/**
 * @brief Destroy the Existence:: Existence object
 *
 */
Existence::~~Existence() {
    numLiving--;
}
```

```
/**
 * @brief Returns the value of numBorn that number of objects of the class that have
 * been instanced.
 *
 * @return unsigned int
 */
unsigned int Existence::getNumBorn() const {
    return numBorn;
}

/**
 * @brief Returns the value of numLiving that total number of objects in existence
 * currently.
 *
 * @return unsigned int
 */
unsigned int Existence::getNumLiving() const {
    return numLiving;
}
```

Address: Istanbul, TURKEY

---

GSM: +90 533 347 6148 | <https://github.com/tamersum11>

c) main.cpp

```
/**
 * @file main.cpp
 * @author tamersum11
 * @brief
 * @version 0.1
 * @date 2022-02-13
 *
 * @copyright Copyright (c) 2022
 *
 */
#include <iostream>
#include <vector>

#include "existence.h"

using namespace std;

Existence chosenOne; //Existence object that is out of society (chosen one)

/**
 * @brief Displays number of born and number of living
 *
 */
void displayHumanity(void) {
    cout << "numBorn    :" << chosenOne.getNumBorn() << endl
         << "NumLiving  :" << chosenOne.getNumLiving() << endl << endl;
}
```

```

/**
 * @brief Creates a Society with the given population
 *
 * @param population population of society
 * @return vector<Existence*>
 */
vector<Existence*> createSociety(unsigned int population) {

    vector<Existence*> v;

    for (int i = 0; i < population; i++)
    {
        v.push_back(new Existence());
    }

    return v;
}

/**
 * @brief 10% of the society is dead
 *
 * @param v
 */
void covid19Pandemic(vector<Existence*> &v) {
    int counter = 0;
    for(auto p : v) {
        if(counter != (int)(v.size() / 10)){
            delete p;
            p = nullptr;
        } else {
            break ;
        }
        counter++;
    }

    v.erase(v.begin(), v.begin() + (int)(v.size() / 10));
}

```

```

/**
 * @brief 50% of the society is dead
 *
 * @param v
 */
void worldWar(vector<Existence*> &v) {
    int counter = 0;

    for(auto p : v) {
        if(counter != (int)(v.size() / 2)){
            delete p;
            p = nullptr;
        } else {
            break ;
        }
        counter++;
    }

    v.erase(v.begin(), v.begin() + (int)(v.size() / 2));
}

/**
 * @brief 100% of the society is dead
 *
 * @param v
 */
void apocalypse(vector<Existence*> &v) {
    for(auto p : v) {
        delete p;
        p = nullptr;
    }

    v.clear();
}

```



```

int main(int argc, char const *argv[])
{
    displayHumanity();

    vector<Existence*> v = createSociety(1000);
    displayHumanity();

    covid19Pandemic(v);
    displayHumanity();

    worldWar(v);
    displayHumanity();

    apocalypse(v);
    displayHumanity();

    return 0;
}

```

d) Output:

```

Starting C:\Users\Tamer\Desktop\Foreks-
C++Questions\Solutions\Question9\build-question9-
Desktop_Qt_6_2_1_MinGW_64_bit-Debug\question9.exe ...
numBorn   :1
NumLiving :1

numBorn   :1001
NumLiving :1001

numBorn   :1001
NumLiving :901

numBorn   :1001
NumLiving :451

numBorn   :1001
NumLiving :1

```

## 10. Question 10

- a) TRUE
- b) TRUE
- c) TRUE
- d) FALSE
- e) TRUE
- f) FALSE
- g) FALSE
- h) TRUE
- i) FALSE
- j) FALSE