Introduction to C++ Programming
Exercises
Set #2

Ali Kashefi
kashefi@stanford.edu

Problem 1. Write the output of the following C++ code in the following box.

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

void f(string A, string &B, const string &C, vector<string> &D){

    string *E = &A ;
    vector<string> *H = &D ;
    B.push_back(E->at(0));
    string *F = &B ;
    string *G = F ;
    (*F).push_back(C[0]) ;
    G->push_back((*H)[D.size()-1].at(D[H->size()-1].length()-1)) ;
    return ;
}

int main(){

    string s1 = "3" ;
    string s2 = "21" ;
    string s3 = "321" ;
    vector<string> s4 = {s2,s3} ;
    f(s1,s2,s3,s4) ;
    cout << s2 ;

    return 0 ;
}
```

your answer:

Problem 2. The intended output of the provided code is 1000, but currently, it results in a compilation error due to its incomplete nature. Your task is to complete the code in a way that it successfully compiles and produces the expected output of 1000. Insert your code within the provided code box at the appropriate location to ensure the integrated code compiles and runs correctly without any errors. To receive credits, you must adhere to the following constraints:

- You are not permitted to delete or modify any existing code, especially within the `structs` and the `main` function.
- Additions to the code are permitted, but they must be outside of the `structs` and the `main` function.

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









struct Mystery{
    private:
        string name = "Mystery";
    public:
        int coin = 100;
        string Name;
        Mystery(string Name) {
        name = Name;
     }
};
```

```
struct Secret {
    private:
        string name = "Secret";
        int coin = 100;
    public:
        Secret(int Coin=10) {
            coin = Coin;
        }
        int getCoin() {
            return coin;
        }
};









Mystery f(Mystery x, Secret y) {
    x.coin = y.getCoin();
    return x;
}









int main() {
    Mystery secret("secret");
    Secret mystery(1000);
    secret = f(mystery, secret);
    cout << secret.coin << endl;
    return 0;
}
```

Problem 3. The code provided below contains a compilation error. Identify the compilation error and write this error in the specified box on the next page. Briefly explain why this error occurs. Provide your explanation in the corresponding box on the next page. As discussed in class, there are two possible ways to resolve this kind of compilation error. Rewrite the code for solution 1 in Box 1 on the following page. Rewrite the code for solution 2 in Box 2 on the subsequent page. While rewriting the code for each solution, you must adhere to the following restrictions:

- Do not change any content inside any of the functions and the `struct`.
- Do not alter the names of any functions or `struct`.
- Do not remove any functions from the code.

Your solutions should correct the identified error while respecting these constraints.

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

struct University {
    private:
        const string name = "UCLA";
        int tuition;
        int population;
    public:
        University(int cost, string Name = "UCLA") {
         population = 10000;}

        void setTuition(int cost) {
            tuition = cost;}

        int getTuition() {
            return tuition;}
};

void g() {
    University UCLA(10000);
    University USC = f(UCLA, "UCLA");
}

University f(University university, string name) {
    University UCLA(10000);
    return UCLA;
}
```

```
int main() {
    University UCLA(10000, "UCLA");
    return 0;
}
```

Write out the specific line of code that is causing the compilation error:

Briefly explain why the error occurs:

Box 1, your code for solution 1:

Box 2, your code for solution 2:

Problem 4. Write the output of the following C++ code in the following box.

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

struct Object{
    private:
        string name;
    public:
        double size;
        bool flag;
        Object(string Name, int Size) {
            name = Name;
            flag = true;
            size = Size;
        }
        int getSize() { return size; }
};

struct Realism {
    private:
        string name = "Realism";
    public:
        vector<Object> reality;
        Realism(Object imagine) {
            reality.push_back(imagine);
        }
};

void f(Object& b, bool flag) {
    if (flag) {
        b.size = 3.0;
        b.flag = false;
    }
    return;
}




Object g(Object b, Object& c, Realism d) {
```

```
    f(c, !d.reality[0].flag);
    b = c;
    return b;
}

int main() {

    Object a1("a1",2.0), a2("a2",4.0), a3("a3",1.0);
    Realism b1(a1), b2(a2), b3(a3);

    vector<Realism> fiction;
    fiction.push_back(b1);
    fiction.push_back(b2);

    a1 = g(fiction[1].reality[0], fiction[0].reality[0], b3);

    cout << a1.getSize() << endl;

    return 0;
}
```

your answer:

Problem 5. In the following code, in the `main` function, the objective is to sequentially invoke two functions, named `action` and `adventure`, using inputs specified within this function (i.e., `"movie"`, `10`, `false`). However, due to security constraints, directly typing the words `action` and `adventure` within the `main` function is prohibited. The implementations of these functions are detailed below the `main` function, but the remainder of the `main` function is inaccessible due to security reasons. Propose a strategy to navigate this challenge and write it briefly in the box on the next page. Note that you are NOT allowed to define a new function in the code. Based on your proposed strategy, write the corresponding code for your solution within the given code. Note that you are allowed to use **a maximum of three semicolons** (`;`) in your written code.

```
// This is the main.cpp file.
#include <iostream>


#include <string>


using namespace std;


void action(string name, int cost, bool flag);


void drama(string title, int value, bool banner);


void adventure(string tag, int price, bool symbol);


int main() {


    string name = "movie";


    int cost = 10;


    bool flag = false;


    return 0;
```

```
}

//******** We are not allowed to see the rest of main.cpp *******//
////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////
```

Briefly explain your strategy:

Problem 6. Matrix multiplication is a fundamental operation in math. The product of two matrices is only possible when the number of columns in the first matrix matches the number of rows in the second matrix. The resulting matrix has the number of rows of the first matrix and the number of columns of the second matrix. For example, let us consider two matrices, A and B, where A is a 3×4 matrix and B is a 4×2 matrix.
Let matrix A (3×4) be:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix},$$

and let matrix B (4×2) be:

$$B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{pmatrix}.$$

The product of A and B is a 3×2 matrix $C = AB$, calculated as follows:

$$C = AB = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{pmatrix},$$

where each element of matrix C is computed as follows:
For c_{11}:

$$c_{11} = a_{11} \cdot b_{11} + a_{12} \cdot b_{21} + a_{13} \cdot b_{31} + a_{14} \cdot b_{41}$$

For c_{12}:

$$c_{12} = a_{11} \cdot b_{12} + a_{12} \cdot b_{22} + a_{13} \cdot b_{32} + a_{14} \cdot b_{42}$$

For c_{21}:

$$c_{21} = a_{21} \cdot b_{11} + a_{22} \cdot b_{21} + a_{23} \cdot b_{31} + a_{24} \cdot b_{41}$$

For c_{22}:

$$c_{22} = a_{21} \cdot b_{12} + a_{22} \cdot b_{22} + a_{23} \cdot b_{32} + a_{24} \cdot b_{42}$$

For c_{31}:

$$c_{31} = a_{31} \cdot b_{11} + a_{32} \cdot b_{21} + a_{33} \cdot b_{31} + a_{34} \cdot b_{41}$$

For c_{32}:

$$c_{32} = a_{31} \cdot b_{12} + a_{32} \cdot b_{22} + a_{33} \cdot b_{32} + a_{34} \cdot b_{42}$$

Now, we want to write a C++ function `product` (provided below) that computes the product of these two matrices and assigns it to the matrix C (i.e., `int C[3][2]`). The function is provided below, however, the code is incomplete and your task is to complete this function. Write your code inside **the following box** (next page). You are allowed to use a maximum of **three semicolons** (;) in your codes. Note that your function should be general for any numbers embedded in matrices A and B; however, you can see an example of these two matrices in `main` function.

```
#include <iostream>
using namespace std;
void product(int A[][4], int B[][2], int row_A = 3, int column_A = 4,
    int row_B = 4, int column_B = 2) {

    int C[3][2] = { {0,0},{0,0},{0,0} };


    for (int i = 0; i < row_A; i++) {


        for (int j = 0; j < column_B; j++){



        }


    }

   cout << C[0][0] << endl;
   cout << C[0][1] << endl;
   cout << C[1][0] << endl;
   cout << C[1][1] << endl;
   cout << C[2][0] << endl;
   cout << C[2][1] << endl;
}

int main() {

    int A[3][4] = {{2,0,2,3},{2,1,2,4},{1,-1,1,-1}};
    int B[4][2] = {{0,4},{4,0},{-2,-2},{1,1}};

    product(A, B, 3, 4, 4, 2);

    return 0 ;}
```

Problem 7. There is a set of five C++ files: `main.cpp`, `player.cpp`, `manager.cpp`, `player.h`, and `manager.h`. These files need to be compiled together to form a cohesive program without any compilation errors. Currently, however, these files are incomplete. Your task is to complete these files. You are not allowed to delete any existing code. You must integrate your code within the existing structure of the provided files. Note that the solution to this question is not unique; however, your code must eventually lead to no compile error.

main.cpp

```
int main() {



return 0;}
```

player.h

```
#ifndef PLAYER_H
#define PLAYER_H
#include <iostream>
#include <string>
using namespace std;

class Player{

    private:

        string name;

        int money;

        bool success;

    public:

        Player(string Name, int Price);

        void setStatus(bool status);

};
```

player.cpp

```
#include "player.h"

Player::Player(string Name, int Price) {

    name = Name;

    money = Price;

}

void Player::setStatus(bool status) {

    success = status;

}

Player::~Player() {

}
```

manager.h

```
#ifndef MANAGER_H
#define MANAGER_H


class Manager {

    private:

        string name;

        string player;

        int money;


    public:

        Manager(int Wealth, string Name = "XYZ");
```

```
        void setPlayer(Player &swimming);

};
```

manager.cpp

```
#include "manager.h"




void Manager::setPlayer(Player& swimming) {

    player = swimming.name;

}
```

Problem 8. Below is a code snippet featuring two incomplete functions, `add` and `print`, along with a provided input-output pair. Your task is to complete these functions within the existing code framework to achieve the specified input-output result. To receive credits, your answer must follow these restrictions: In function `print`, you are limited to using only one semicolon (;), and in function `add`, you are allowed to use a maximum of two semicolons (;). You are not permitted to use and write the `cout`, or any other similar printing functions. You are not permitted to include any other libraries.

Sample input:

```
50
```

Sample output:

```
20
30
100
40
50
```

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

struct Home{
    private:
        int value;
    public:
        Home* right = nullptr;
        Home(int cost) {value = cost;}
        int getValue() {return value;}
};

void print(Home starting_home) {


    while (starting_home.right != nullptr) {


        cout << starting_home.right->getValue() << endl;


    }

    return;
```

```
}

void add(Home& current, Home& before, Home& after) {




    return;
}

int main() {
    int n;
    cin >> n;
    Home B1(10), B2(20), B3(30), B4(40), B5(n);

    Home* ptr_B2 = &B2;
    B1.right = ptr_B2;

    Home* ptr_B3 = &B3;
    B2.right = ptr_B3;

    Home* ptr_B4 = &B4;
    B3.right = ptr_B4;

    Home* ptr_B5 = &B5;
    B4.right = ptr_B5;

    Home A1(100);

    add(A1, B3, B4);

    print(B1);

    return 0;
}
```

Problem 9. Let us consider a pointer that addresses another pointer. For example, in the following code, in the `main` function, the pointer `c` is designed to address the pointer `b`. Moreover, the output of the following code is 5.

```
#include <iostream>
using namespace std;
int main() {
    int a = 5;
    int* b = &a;
    int** c = &b;
    cout << c << endl;
    return 0;}
```

Now, write the output of the following C++ code in the following box. To receive credits, you must show your work.

```
#include <iostream>
using namespace std;
int main() {
    double number = 2.0;
    double price = -1.0;
    double *value = &number;
    *value *= 2.0;
    price -= *value;
    *value += number;
    double* cost = &number;
    *cost *= *value;
    double** charge = &value;
    value = &price;
    **charge = *value + *cost - **charge;
    *value = 2.0;

    cout << number + price << endl;

    return 0;}
```

your answer:

Problem 10. The intended output of the provided code is `abcdE`, but currently, it results in a compilation error due to its incomplete nature. Your task is to complete the code in a way that successfully compiles and produces the expected output of `abcdE`. Insert your code within the provided code box at the appropriate location to ensure the integrated code compiles and runs correctly without any errors. To receive credits, you must adhere to the following constraints:

- You are not permitted to delete or modify any existing code, especially within the `main` function.
- Additions to the code are permitted, but they must be outside of the `main` function.

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




string g(string& x) {
    return x + "E";
}




int main() {
    cout << g("abcd");
    return 0;
}
```

Problem 11. Write a C++ program, **in the box provided on the next page**, that takes two `string`s as input, each representing a positive integer number, and outputs the sum of these integers as an `int`. We assume that the input strings do not start with a '0'.

You are allowed only to use the `<iostream>`, `<string>`, and `<cmath>` libraries. You are NOT allowed to use the `<vector>` library, arrays, the `<stack>` library, or built-in functions for converting a `string` to an `int` and vice versa. As the concept of consecutive ASCII values has not been covered in this class, you are also NOT allowed to use it. Non-compliance with any of these policies will result in zero credit for this problem.

To help you, we provided a few lines of code in the box.

Sample Input:

```
3

8
```

Sample Output:

```
11
```

Sample Input:

```
24

3
```

Sample Output:

```
27
```

Sample Input:

```
2024

145667
```

Sample Output:

```
147691
```

your answer:

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

int main() {
    string s1, s2;
    cin >> s1 >> s2;

    return 0;
}
```

Problem 12. The code provided below contains a compilation error. Identify the compilation error and write this error in the specified box on the next page (1 point). Briefly explain why this error occurs. Provide your explanation in the corresponding box on the next page (1 point). As discussed in class, there are two possible ways to resolve this kind of compilation error. Rewrite the code for solution 1 in Box 1 on the following page (2 points). Rewrite the code for solution 2 in Box 2 on the subsequent page (2 points). While rewriting the code for each solution, you must adhere to the following restrictions:

- Do not change any content inside any of the functions.
- Do not alter the names of any functions.
- Do not remove any functions from the code.

Your solutions should correct the identified error while respecting these constraints.

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

void g(string y) {
    cout << f(y, 0);
    return;
}

char f(string name, int x) {
    name += "UCLA";
    return name[x];
}

int main() {
    double y1, y2;
    return 0;
}
```

Write out the specific line of code that is causing the compilation error:

Briefly explain why the error occurs:

Box 1, your code for solution 1:

Box 2, your code for solution 2: