Name : Malaviya Parth Mahendrabhai

Roll no : CE025

Id : 24CEUOG067

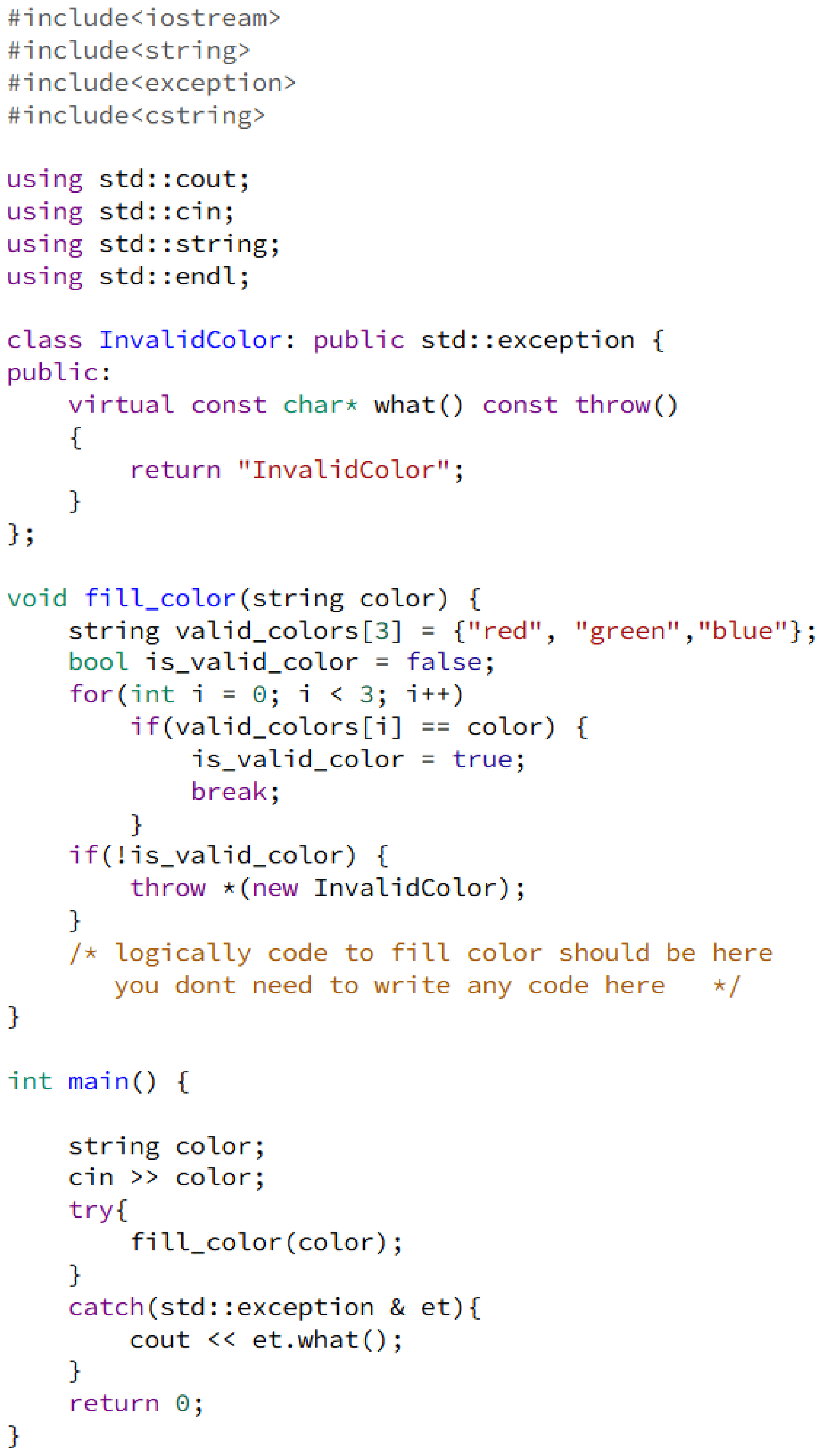
Division : A1

Batch : 2024 - 28

Subject : PPS 2

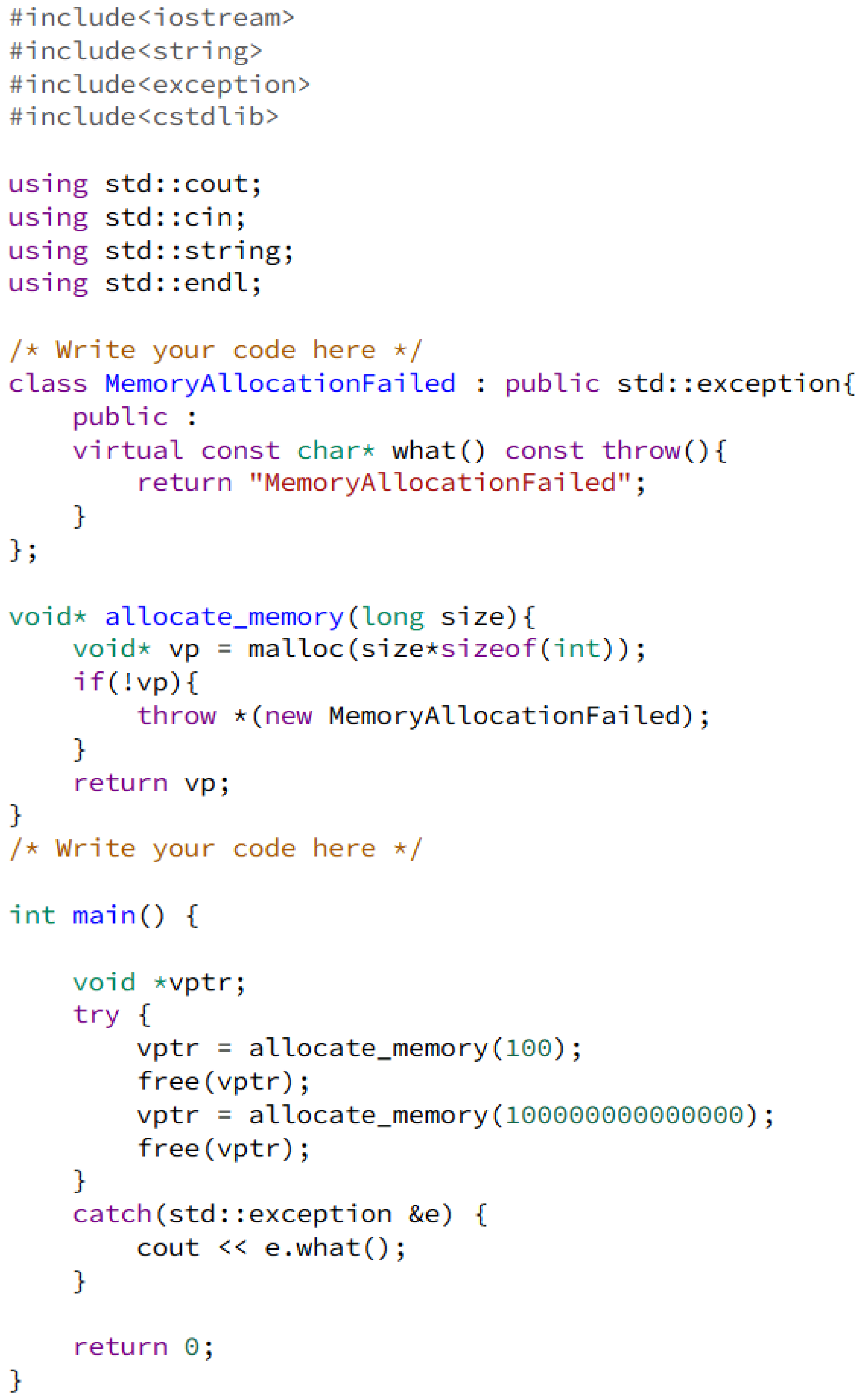
Lab : 12

Program : 1



```cpp
#include<iostream>
#include<string>
#include<exception>
#include<cstring>

using std::cout;
using std::cin;
using std::string;
using std::endl;

class InvalidColor: public std::exception {
public:
    virtual const char* what() const throw()
    {
        return "InvalidColor";
    }
};

void fill_color(string color) {
    string valid_colors[3] = {"red", "green","blue"};
    bool is_valid_color = false;
    for(int i = 0; i < 3; i++)
        if(valid_colors[i] == color) {
            is_valid_color = true;
            break;
        }
    if(!is_valid_color) {
        throw *(new InvalidColor);
    }
    /* logically code to fill color should be here
       you dont need to write any code here    */
}

int main() {

    string color;
    cin >> color;
    try{
        fill_color(color);
    }
    catch(std::exception & et){
        cout << et.what();
    }
    return 0;
}
```

Program : 2

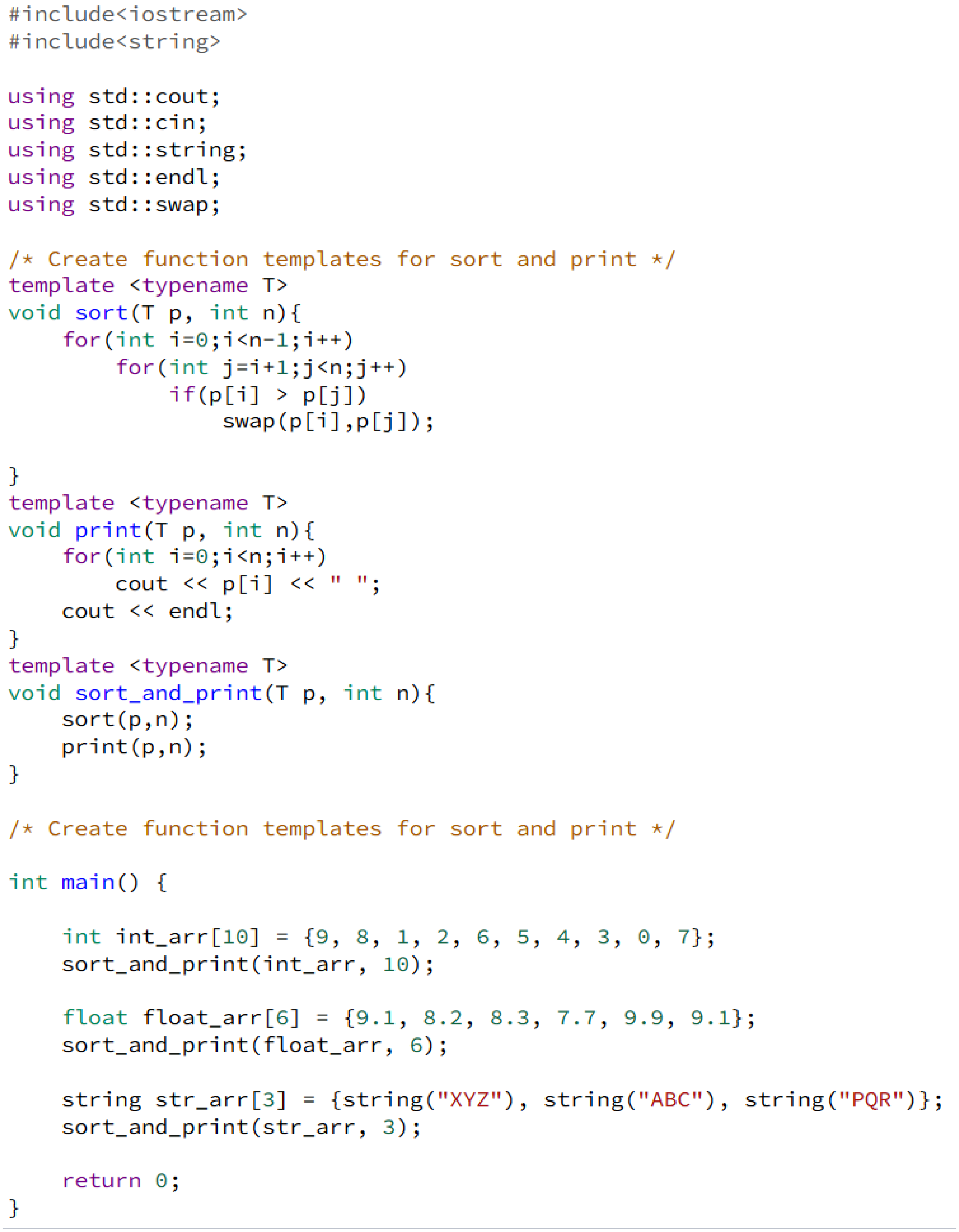


```
#include<iostream>
#include<string>
#include<exception>
#include<cstdlib>

using std::cout;
using std::cin;
using std::string;
using std::endl;

/* Write your code here */
class MemoryAllocationFailed : public std::exception{
    public :
    virtual const char* what() const throw(){
        return "MemoryAllocationFailed";
    }
};

void* allocate_memory(long size){
    void* vp = malloc(size*sizeof(int));
    if(!vp){
        throw *(new MemoryAllocationFailed);
    }
    return vp;
}
/* Write your code here */

int main() {

    void *vptr;
    try {
        vptr = allocate_memory(100);
        free(vptr);
        vptr = allocate_memory(100000000000000);
        free(vptr);
    }
    catch(std::exception &e) {
        cout << e.what();
    }

    return 0;
}
```

Program : 3

```
#include<iostream>
#include<string>

using std::cout;
using std::cin;
using std::string;
using std::endl;
using std::swap;

/* Create function templates for sort and print */
template <typename T>
void sort(T p, int n){
    for(int i=0;i<n-1;i++)
        for(int j=i+1;j<n;j++)
            if(p[i] > p[j]){
                swap(p[i],p[j]);
            }
}
template <typename T>
void print(T p, int n){
    for(int i=0;i<n;i++)
        cout << p[i] << " ";
    cout << endl;
}
/* Create function templates for sort and print */

int main() {

    int int_arr[10] = {9, 8, 1, 2, 6, 5, 4, 3, 0, 7};
    sort(int_arr, 10);
    print(int_arr, 10);

    float float_arr[6] = {9.1, 8.2, 8.3, 7.7, 9.9, 9.1};
    sort(float_arr, 6);
    print(float_arr, 6);

    string str_arr[3] = {string("XYZ"), string("ABC"), string("PQR")};
    sort(str_arr, 3);
    print(str_arr, 3);

    return 0;
}
```

Program : 4



```
#include<iostream>
#include<string>

using std::cout;
using std::cin;
using std::string;
using std::endl;
using std::swap;

/* Create function templates for sort and print */
template <typename T>
void sort(T p, int n){
    for(int i=0;i<n-1;i++)
        for(int j=i+1;j<n;j++)
            if(p[i] > p[j])
                swap(p[i],p[j]);

}
template <typename T>
void print(T p, int n){
    for(int i=0;i<n;i++)
        cout << p[i] << " ";
    cout << endl;
}
template <typename T>
void sort_and_print(T p, int n){
    sort(p,n);
    print(p,n);
}

/* Create function templates for sort and print */

int main() {

    int int_arr[10] = {9, 8, 1, 2, 6, 5, 4, 3, 0, 7};
    sort_and_print(int_arr, 10);

    float float_arr[6] = {9.1, 8.2, 8.3, 7.7, 9.9, 9.1};
    sort_and_print(float_arr, 6);

    string str_arr[3] = {string("XYZ"), string("ABC"), string("PQR")};
    sort_and_print(str_arr, 3);

    return 0;
}
```