

1. The following program contains one function call that won't work properly. Cross it out (don't make any other corrections to the code). Show what will be displayed when only correct function calls are invoked:

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

```
void f(char a, string s, float x = 3.14);
```

```
void f(string w = "No", int n = 3);
```

	OUTPUT:
int main()	
{ f();	
f('Y', 2.71);	
f("Y", "Hello");	
f("CS2260");	
return 0;	
}	

```
void f(char a, string s, float x)
```

```
{  cout << x << " ", " << s << " ", " << a << endl; }
```

```
void f(string w, int n)
```

```
{  cout << n << " ", " << w << endl; }
```

2. Show the output of the following code fragments:

(a)

```
string s1 = "nine";
string s2 = "eight";
cout << s1.find_first_of(s2);
cout << endl;
cout << s2.find_first_not_of(s1);
cout << endl;
```

(b)

```
string s1="AB";
string s2="abcdef";
s2.insert(3, s1);
cout << s2 << endl;
s2.replace(2,2, s1);
cout << s2 << endl;
```

(c)

```
string s1 = "display";
string s2 = "dispatch";
if(s1<=s2)
  cout << "yes" << endl;
else
  cout << "no" << endl;
s1.erase(1, 4);
cout << s1 << endl;
```

3. Given the following class declaration:

```
class Quest{
private: int a;
public: int geta()
      {return a;}
      . . .
};
```

The top-level function `show` is supposed to display the value of the private data member `a` of an object of the class `Quest` passed to this function. Fill in the table below for: (A) pass by value; (B) pass by address; (C) pass by reference. For the rightmost column, assume that the object `Q` of the class `Quest` was defined and initialized properly

	Function prototype	Function definition	Function call
(A)			
(B)			
(C)			

4. Correct the errors in the following code:

```

#include <iostream>
using namespace std;
class test
{   int *pi;
public:
    test(){pi = new int;}
    test(int n){pi = new int; pi = n;}
    int get_pi(){ return *pi;}
    void set_pi(){*pi = n;}
    ~test(){ delete [] pi;}
};
void display(test x){ cout << x.pi << endl;}
int main()
{
    test a(10);
    cout << * a.pi << endl;
    test b;
    b.set_pi(15);
    b.display();
    return 0;
}

```

Correction:

5. Show the output of the following program:

```
#include <iostream>
#include<string>
using namespace std;
class Student
{public:
    Student()
    {   status = "freshman";
        age = 19;
        cout << "Constructor1" << endl;
    }
    Student(int i)
    {   status = "senior";
        age = i;
        cout << "Constructor2" << endl;
    }
    Student(const Student & a)
    {   status = a.status;
        age = a.age + 5;
        cout << "Constructor3" << endl;
    }
    ~Student(){cout << "Destructor" << endl;}
    void show();
private:
    string status;
    int age;
};
void Student::show()
{ cout << "Status: << status;
  cout << ", age:" << age << endl;
}
void display1(Student a)
{   a.show(); }
void display2(Student & a)
{   a.show(); }
int main()
{   Student st1(19);
    display2 (st1);
    Student st2, st3(st1);
    display2(st2);
    display2(st3)
    st2 = st1;
    display1(st2);
    return 0;
}
```

O U T P U T :

6. Below is an unfinished declaration of the class named `Length`. Complete the declaration of this class so that: (a) private members `feet` and `inches` will be initialized to the values provided when an object is defined; (b) it will not be allowed to define objects of the class `Length` without any initial values; (c) there will be a public method `conversion` converting length in feet and inches to centimeters (1 foot=30.48 cm; 1 inch=2.54 cm); d) there will be a public method `show` displaying a message like this: 3 feet and 5 inches equal 104.14 cm (of course, instead of displaying constant numbers you must display values of corresponding variables!); e) provide interface functions for setting and getting values of the private members; f) write a method for adding two objects of this class, so that in the sum members `feet` and `inches` will be sums of corresponding private members of the objects to be added; if the resulting value of `inches` is greater than 11, take an appropriate action.

```
class Length{
private:
    int feet, inches;
    float cm;
    //is anything else necessary?

public:
    //add necessary functions
```

Please, close the class declaration properly!

7. Write a test driver to demonstrate public functions of the class Length .

8. Show the output of the following program:

```
#include <iostream>
using namespace std;
namespace one
{
    int a, b;
}
int main()
{
    one::a=6;
    one::b=1;
    {
        int b=5, a=3;
        cout << a << "; " ;
        cout << one::b << endl;
        one::a=a;
    }
    using namespace one;
    cout << a << "; " << b << "; ";
    cout << one::a << endl;
    return 0;
}
```

Output: