Ghulam Ishaq Khan Institute (GIKI)

Assignment # 2	
Subject: Object Oriented Programming & Design	**Course Code:** CS - 112 - A, F - Spring - 24
Class: BS AI & BS ME– 2nd, **Batch:** 33 (Fall-23-Intake)	**Deadline:** 17/March/2024 - Sunday (11:59 - PM)
Course Instructor: M. Qasim Riaz - Lecturer - FCSE	**Total Marks:** 30 (Marks are divided problem wise)
Course TA's: Mr. Abubakar (BS-ME), Miss Munazza (BS-AI)	
Name:	**Reg #:**

MID TERM EXAM (SAMPLE) PRACTICE ASSIGNMENT

INSTRUCTIONS

1. Print the complete given paper and solve by hand in given space.
2. Mention your reg number and name at bottom of each page.
3. Take picture & upload the completed assignment in PDF format.
4. If above requirements are not met, assignment will not be marked.

Question # 1 (10 Marks)

1.1 Differentiate between shallow copy and deep copy. In which cases shallow copy will suit and when do we require deep copy? Explain in 3-4 sentences. 3

Shallow Copy: Both objects data members are pointing to same direction memory location therefore they have same values. It is found by default copy constructor

Deep Copy: Both objects data members have their own copies of the values in the memory. It is found by used defined copy constructor. Shallow copy is useful for non-dynamic memory while deep copy is useful for dynamic.

1.2 Understand the following code. You need to provide source code for a copy constrcutor that performs deep copy. 7

```
#include <iostream>
using namespace std;
class Shape {
   public:
      string name;
      int *ptr;
      Shape( string n, int len );
};

Shape::Shape(string n, int len) {
   cout << "Normal constructor" << endl;
   name = n;
   ptr = new int;
   *ptr = len;
}

int main() {
  Shape s1("Line", 10);
  Shape s2 = s1;
  cout << "Length of Shape " << s1.name << " is "<<*s1.ptr <<endl;
  cout << "Length of Shape " << s2.name << " is "<<*s2.ptr <<endl;
  return 0;
}
```

Write Your Code Here

```
#include <iostream>
using namespace std;
class Shape
{
public:
string name;
int *Ptr;
Shape (string n, int len)
{ cout << "Normal constructor"<<endl;
name = n
Ptr = new int;
*ptr = len;
}
Shape ( const Shape &obj)
{ cout << "copy constructor"<<endl;
name = obj.name;
Ptr = new int;
*Ptr = *(obj.Ptr);
```

```
int main()
{
    Shape S1("line", 10);
    Shape S2 = S1;
    cout<<"length of shape"<<S1.name<<"*(S1.Ptr)<<endl;
    cout<<"length of shape"<<S2.name<<"is"<<(S2.Ptr)<<endl;
    return 0;
}
```

Question # 2 (10 Marks)

2.1 Generally, we can say that we should use "friend function/class" only when it is really necessary. That means it is desirable to minimize the usage of "friend function/class". Why? Explain in detail 4

using friend function or classes existing can lead to problems such as encapsulations violation increased coupling, working, difficulty in testing and debugging and securing risks it's generally, it's better to rely on well designed interfaces. and maintain encapsulation for code clarity.

2.2 Trace through the below mentioned code. What's the output of the code? In case error, explain the reason and re-write the correct version of the code. 6

```
#include <iostream>
using namespace std;
class Car {
    int length, width, height;
public:
    Car(int x, int y, int z)
    {
        length = x;
        width = y;
        height = z;
    }
};
class Truck {
    int length, width, height;
    friend class Car;
public:
    void twice(Car a)
    {
        length = a.length*2;
        width = a.width*2;
        height = a.height*2;
        cout<<"Twice the size: l="<<length<<" w = "<<width<<" h = "<<height;
    }
};

int main() {
    Car car(3, 2, 2);
    Truck truck;
    truck.twice(car);
}
```

The error lies in the truck class, truck cannot access the private variables of class. For code to run, we have to make it Public.

```
#include<iostream>
using namespace std;
class Car {
public:
    int length, width, height;
    // remaining code.
```

Reg #: 202353, Name: M. Umar

Question # 3 (10 Marks)

3.1 Differentiate between static data member and constant data member of a class. 3

Static data members are stored array all instances of class and have a single copy to entire class. constant data member are unique to each object and represent values that can not be modified.

3.2 What will be the output of the following program? Explain the reason for each output statement. 7

```
#include <iostream>
using namespace std;
void StaticTest(){

    static int x = 1;
     x = ++x;

    int y = 1;
    y = ++y;
    cout<<"x = "<<x<<endl;
    cout<<"y = "<<y<<endl;
}
int main()
{
    StaticTest();
    StaticTest();
    return 0;
}
```

x = 2
y = 2
x = 3
y = 2

As x is static it means that it is stored in static memory rather than heap or stack while y is not with static keyword and it will be reinitialized to 1.

********** End of Examination Paper **********