OOP-LAB Task 5

Roll No: 24P-0706 **Dept: BS-CS**

Name: Aazan Noor Khuwaja **Section : 2D**

QNS-1:

Code: By Using Deep Copy

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

class english {
   char *sentence;
   int size;
public:
   english() : size(0) {
      sentence = new char[1];
      sentence[0] = '\0';
   }
   english(const char n[], int s) : size(s) {
      sentence = new char[size + 1];
      strcpy(sentence, n);
   }
   english(const english &e) {
      cout<<"Copy constructor is called"<<endl;
      size = e.size;
      sentence = new char[size + 1];
```

```
        strcpy(sentence, e.sentence);
    }
    ~english() {
        delete[] sentence;
    }
    void show() {
        cout << "Name: " << sentence << endl;
        cout << " Size: " << size << endl;
    }
};

int main() {
    english e1("Aazan", 5);
    e1.show();
    english e2(e1);
    e2.show();
    english e3(e2);
    e3.show();
    return 0;
}
```

Output:

```
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-5> g++ -g task1.cpp -o ./output
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-5> ./output
Name: Aazan
 Size: 5
Copy constructor is called
Name: Aazan
 Size: 5
Copy constructor is called
Name: Aazan
 Size: 5
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-5>
```

QNS-1:

Code: By Using Shallow Copy

```
#include<iostream>
#include<string>
#include<cstring>
using namespace std;
class english{
   string sent_ence;
   int si_ze;
public:
   english() : si_ze(0) {
      sent_ence = "";
   }
   english(string n){
      sent_ence = n;
      si_ze = strlen(n.c_str());
   }
   english(const english &e) {
      cout<<"Copy constructor is called"<<endl;
      si_ze = e.si_ze;
      sent_ence = e.sent_ence;
   }
   void show() {
      cout << "Name: " << sent_ence << endl;
      cout << " Size: " << si_ze << endl;
   }

};
int main() {
```

```
    english e1("Aazan");
    e1.show();
    english e2(e1);
    e2.show();
    english e3(e2);
    e3.show();
    return 0;
}
```

Output:

```
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-5> g++ -g task.11.cpp -o ./output
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-5> ./output
Name: Aazan
 Size: 5
Copy constructor is called
Name: Aazan
 Size: 5
Copy constructor is called
Name: Aazan
 Size: 5
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-5>
```

QNS-2:

Code:

```
#include<iostream>
using namespace std;
class serialnumber{
    int serial_number;
    static int count_objects ;
    public:
    serialnumber()
   {
```

```
        count_objects++;
        serial_number=count_objects;
        cout<<"object "<<count_objects<<" created!"<<endl;
    }
    static int getfun()
    {
     return count_objects;
    }
    void report_serial_numb()
    {
     cout<<"Serial Number is:"<<serial_number<<endl;

    }

};
    int serialnumber::count_objects=0;
     int main()
{
     serialnumber s1;
     s1.report_serial_numb();
     serialnumber s2;
     s2.report_serial_numb();
     serialnumber s3;
     s3.report_serial_numb();



     cout<<"The Total objects are "<<serialnumber::getfun()<<endl;
```

```
    return 0;


}
```

Output:

```
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-5> g++ -g task2.cpp -o ./output
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-5> ./output
object 1 created!
Serial Number is:1
object 2 created!
Serial Number is:2
object 3 created!
Serial Number is:3
The Total objects are 3
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-5>
```

QNS-3:

Code:

```
#include<iostream>

using namespace std;

class time{

   int hours;

   int minutes;

   int seconds;

   public:

   time():hours(0),minutes(0),seconds(0){};

   time(int hours,int minutes, int seconds){

      this->hours=hours;

      this->minutes=minutes;

      this->seconds=seconds;

   }

   void display()

   {

      cout<<"Time  "<<hours<<":"<<minutes<<":"<<seconds<<endl;
```

```
    }
    time addtwoobjects(time t1,time t2)
    {
        time store;
        store.hours=t1.hours+t2.hours;
        store.minutes=t1.minutes+t2.minutes;
        store.seconds=t1.seconds+t2.seconds;
        if(store.seconds>=60)
        {
            store.seconds-=60;
            store.minutes++;

        }
       if(store.minutes>=60)
       {
            store.minutes-=60;
            store.hours++;

       }
       return store;
    }

};
int main()
    {
        time t1(2,33,50);
        time t2(12,24,17);
        cout <<"Before Adding Objects of time :"<<endl;
        t1.display();
```

```
    t2.display();
    time t3,t4;
    t4=t3.addtwoobjects(t1,t2);
    cout <<"After Adding Objects of time :"<<endl;
    t4.display();
    return 0;
}
```

Output: