Name:- Murali krishna CSE 202

Section:- K19FR Assignment

RollNO:-01

RegNO:-11907747

1) C++ Program to count number of words lines & Total size of a Text file.

```
# include <iostream>
# include <fstream>
using namespace std;
int main()
{
ifstream fin("Text.text");
int line=1, word=1;
char ch;
int size;
fin.seekg (0, ios::end);
size = fin.tellg();
fin.seekg (0, ios::beg);
while (fin)
{
 fin.get (ch);
 if (ch==" "||ch=='n')
word++;
if (ch='n')
line++;
```

```
}
cout<<"\nword = "<<word<<"\n lines =
    <<lines<<"\n size = "<<size<<"\n";
fin.close();
return 0;
}
```

---

2) Program to implement Hybrid inheritance

```
#include<iostream>
#include<conio.h>
using namespace std;
class student
{
Protected:
    int rollno;
Public
    void getdata(int a)
  {
    rollno=a;
    cout<<"Enter the rollno";
    cin>>a;
  }
    void putdata()
    {
     cout<<"\n rollno"<<rollno;
    }
  };
class test:public student
{
  Protected:
     float part1, Part2
Public:
  void getmarks(float x, float y)
{
  part1=x;
  part2=y;
}
void putmarks()
{
 cout<<"part1 = "<<part1<<
"\npart2 = "<<part2;
}
};
class sports
{
  Protected:
    float score;
Public
  void getscore(float s)
  {
    score=s
  }
```

```
void putscore ( )
{ cout<<"\n sports wt " << score ;
}
};
class result : Public test . public sports
{ float total;
  Public :
    void display ( );
  { void result : display( );
  }
  total = part1 + part2 + score
    put data ( );
    put marks ( );
    put score ( );
  cout << " total score" << total;
  };
int main ( )
{ result student ;
student. get data (2345);
 student . get marks (89 , 90 ),
student . get score (96 );
student . display ( );
getch( );
return 0 ;
}
```

3) c++ Program to add two objects using binary operator overloading

```
#include <iostream>
using namespace std;
class complex
{
 float x, y;
public:
complex (float real, float imag)
{ x = real;
  y = imag;
}
complex operator+ (complex)
{ complex temp;
 temp.x = x + c.x;
 temp.y = y + c.y;
return temp;
}
void display (void);
{
 cout << x << " " << y << endl;
}
};
int main()
{
 complex c1, c2, c3
c1 = complex (4.5, 8.5);
c2 = complex (9.6, 8.9)
c3 = c1 + c2;
cout << "c1 = ";
c1.display();
cout << "c2 = ";
c2.display();
cout << "c3 = ";
c3.display();
return 0;
}
```

4) write a programm on virtual function

```
# include <iostream>
using name space
class A
  {
   Public:
   virtual void display()
    {
     cout<<" Base class is involved "<<endl;
    }
};
class B: public A
{
  Public:
  void display()
  {
   cout<<"Derived class is involved "<<endl;
  }
};
int main()
{
  ~~class A a~~
   A* a;
   B* b;
   a = &b;
   a -> display();
  }
```

output

Derive class is involved