BAPATLA ENGINEERING COLLEGE, BAPATLA
(AUTONOMOUS)

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS ENGINEERING

LAB CODE: EC 362

NAME OF THE LAB: OBJECT ORIENTED PROGRAMMING USING C++

2012-2013

Prepared by

CHESTI ALTAFF HUSSAIN M.Tech
Lecturer
Department of ECE

List of Lab Programs

Write C++ Programs to illustrate the concept o the following

1. Arrays
2. Structures
3. Pointers
4. Objects and Classes
5. Console I/O operations
6. Scope Resolution and Memory Management Operators
7. Inheritance
8. Polymorphism
9. Virtual Functions
10. Friend Functions
11. Operator Overloading
12. Function Overloading
13. Constructors and Destructors
14. *this* Pointer
15. File I/O Operations

1. C++ Program on Arrays

Program: Write a C++ Program to display names, roll no's, and grades of 3 students who have appeared in the examination. Declare the class of name, roll no's and grade. Create an array of class objects. Read and display the contents of the array.

```
#include <iostream.h>
#include<conio.h>
void main()
{
int k=0;
class stud
{
public:
char name[12];
int rollno;
char grade[2];
};
class stud st[3];
while(k<3)
{
clrscr();
gotoxy(2,4);
cout<<"Name :  ";
gotoxy(17,4);
cin>> st[k].name;
gotoxy(2,5);
cout<<"Roll No. : ";
```

```
gotoxy(17,5);
cin>>st[k].rollno;
gotoxy(2,6);
cout<<"Grade:  ";
gotoxy(17,6);
cin>>st[k].grade;
st[k].grade[1]='\0';
puts("press any key…");
getch();
k++;
}
k=0;
clrscr();
cout<<"\n Name\t Rollno \t Grade\n";
while(k<3)
{
cout<<st[k].name<<"\t"<<st[k].rollno<<"\t"<<st[k].grade<<"\n";
k++;
}
}
```

2. C++ Program on Structures

Program: Write a C++ program to declare *struct*. Initialize and display contents of member variables.

```
#include<iostream.h>
#include<constream.h>

struct item
{
int codeno;
float prize;
int qty;
};

void main()
{

item a,*b;
clrscr();

a.codeno=123;
a.prize=150.75;
a.qty=150;

cout<<"\n With simple variable";
cout<"\n Codeno: "<<a.codeno;
cout<<"\n Prize:"<<a.prize;
cout<<"\n Qty:"<<a.qty;

b->codeno=124;
b->prize=200.75;
b->qty=75;

cout<<"\n with pointer to structure";
cout<<"\n codeno:"<<b->codeno;
cout<<"\n prize"<<b->prize;
cout<<"\n qty:"<<b->qty;

}
```

3. C++ Program on Pointers

Program: Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member.

```
#include<iostream.h>
#include<conio.h>
void main()
{

class man
{
public:
char name[10];
int age;
};

man m={“RAVINDRA”, 15};
man *ptr;

ptr=&(man)m;

clrscr();

cout<<”\n”<<m.name<<” “ <<m.age;
cout<<”\n”<<ptr->name<<” “<<ptr->age;
}
```

4. C++ Program on Objects and Classes

Program: Given that an EMPLOYEE class contains following members:
data members: Employee number, Employee name, Basic, DA, IT, Net Salary and print data members. Write a C++ program to read the data of N employee and compute Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the gross salary) .

```
#include <iostream.h>
 #include <conio.h>

class employee
 {
 char name[10];
int no;
float basic;
 float da;
 float it;
 float ns;
 float gs;

 public:
 void input()
{
 cout <<"Enter number:";
cin >> no;
cout <<"Enter name:";
cin >> name;
cout <<"Enter salary:";
cin >> basic;
}

 void calculate()
 {
 da = 0.52 * basic;
gs = da + basic;
it = 0.3 * gs;
ns = gs - it;
 }

 void output()
```

```
 {
 cout<<no<<'\t'<<name<<'\t'<<basic<<'\t'<<ns<<'\t'<<gs <<'\n';
}
 };

void main()
{
employee emp[20];
 int n,i;
clrscr();
cout << "Enter no of employees:";
cin >> n;
for(i=0;i<n;i++)
{
 emp[i].input();
emp[i].calculate();
}
cout<<"NUMBER"<<'\t'<<"NAME"<<'\t'<<"BASIC"<<'\t'<<"NET"<<'\t'
<<"GROSS" << "\n";

 for(i=0;i<n;i++)
 {
emp[i].output();
}
getch();
}
```

5. C++ Program on Console I/O operations

Program: Write a C++ to illustrate the concepts of console I/O operations.

```
#include<iostream.h>
#include<conio.h>

void main()
{
cout.width(5);
cout<<”A”;
cout.width(15);
cout<<”B”;
cout.precision(2);
cout<<3.1452;
cout.fill(‘/’);
cout.width(20);
cout<<”WEL”<<endl;
cout.fill(‘-‘);
cout.width(10);
cout<<”DONE”;
}
```

6. C++ Program on Scope resolution and Memory Management Operators

6.1 Program: Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.

```
#include<iostream.h>
#iclude<conio.h>
int a = 10;

main()
{
clrscr();
int a=20;
cout<<"::a="<<::a;
cout<<" a="<<a;
return 0;
}
```

6.2 Program: Write a C++ program to allocate memory using *new* operator.

```
#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();

int *p= new int[3],k;

for(k=0;k<3;k++)
{
cout<<"\n Enter a number: ";
cin>>*p;
p++;
}

p- = 3;

cout<<"\n Entered numbers with their address are: \n";
```

```
for(k=0;k<3;k++)
{
cout<<"\n\t"<<*p<<"\t"<<(unsigned)p;
p++;
}
p- = 3;

delete p;
}
```

7. C++ Program on Inheritance

Program: Write a C++ program to create multilevel inheritance. Create classes A1,A2, A3.

```
#include<iostream.h>
#include<constream.h>

class A1
{
protected:
char name[15];
int age;
};

class A2 : public A1
{
protected:
float height,weight;
};

class A3: public A2
{
protected :

char sex;
public:

void get()
{
cout<<"Name : ";  cin>>name;
cout<<"Age : ";  cin>>age;
cout<<"Sex : ";  cin>>sex;
cout<<"Height : ";  cin>>height;
cout<<"Weight : ";  cin>>weight;
}

void show()
{
cout<<"\nName : "<<name;
```

```
cout<<"\nAge : "<<age<<"Years";
cout<<"\nHeight : "<<Height<<"Feets";
cout<<"\nSex : "<<sex;
cout<<"\n Weight : "<<weight<<"kg.";
}
};

void main()
{
clrscr();
A3 x;
x.get();
x.show();
}
```

8. C++ Program on Polymorphism

Program: Write a C++ program to create an array of pointers. Invoke functions using array objects.

```
#include<iostream.h>
#include<constream.h>

class A
{
Public:
virtual void show()
{
cout<<”A\n”;
}
};

class B:public A
{
public:
void show()
{
cout<<”B\n”;
}
};

class C:public A
{
public:
void show();
{
cout<<”C\n”;
}
};

class D:public A
{
public:
void show()
{
```

```
cout<<”D\n”;
}
};

class E:public A
{
public:
void show()
{
void show()
{
cout<<”E”;
}
};

void main()
{
clrscr();
A a;
B b;
C c;
D d;
E e;

A *pa[] = { &a,&b,&c,&d,&e};

for(int j=0;j<5;j++)
pa[j]->show();
}
```

9. C++ Program on Virtual Functions

Program : Write a C++ program to use pointer for both base and derived classes and call the member function. Use Virtual keyword.

```
#include<iostream.h>
#include<conio.h>

class super
{
public:

virtual void display()
{
cout<<"\n In function display() class super";
}

virtual void show()
{
cout<<"\n In function show() class super";
}
};

class sub: public super
{
public:

void display()
{
cout<<"\n In function display() class sub";
}

void show()
{
cout<<"\n In function show() class sub";
}
};

int main()
{
```

```
clrscr();
super S;
sub A;
super *point;
cout<<"\n Pointer point points to class super\n";
point= &s;
point->display();
point->show();
cout<<"\n\n Now pointer point points to derived class sub\n";
point= &A;
point->display();
point->show();
return 0;
}
```

10.C++ Program on Friend Functions

Program: Write a C++ program to overload unary operator using *friend* function.

```
#include<iostream.h>
#include<constream.h>

class complex
{
float real,imag;

public:

complex()
{
real=imag=0;
}

complex(float r,float i)
{
real = r;
imag = i;
}

friend complex operator – (complex c)
{
c.real = -c.real;
c.imag = -c.imag;
return c;
}

void display()
{
cout<<"\n Real:"<<real;
cout<<"\n Imag: "<<imag;
}
};

void main()
```

```
{
clrscr();

complex c1(1.5,2.5),c2;
c1.display();
c2=-c1;
cout<<"\n\n After Negation \n");
c2.display();
}
```

11.C++ Program on Operator Overloading

Program: Write a C++ program to overload – operator.

```
#include<iostream.h>
#include<conio.h>

class num
{
private:
int a,b,c,d;
public:

num(int x,int y,int z,in w)
{
a=x;
b=y;
c=z;
d=w;
}

void show(void);
void operator –();
};

void num :: show()
{
cout<<”A= “”<<a<<”B= “”<<b<<”C “”<<c<<”D “”<<d;
}

void num :: operator –()
{
a= -a;
b= -b;
c= -c;
d= -d;
}

main()
{
```

```
clrscr();
num(X(2,2,8,4);

cout<<"\n Before negation of X:  ";
X.show();
-X;
cout<<"\n After negation of X : ";
X.show();
return 0;
}
```

12.C++ Program on Function Overloading

Program: Write a C++ program to create a class called COMPLEX and implement the following overloading functions ADD that return a COMPLEX number. I. ADD (a, s2) - where a is an integer (real part) and s2 is a complex number. II. ADD (s1, s2)-where s1 & s2 are complex numbers.

```
#include <iostream.h>
 #include <conio.h>
#include <math.h>

class complex
{
int real;
int img;

public:

 void input()
 {
 cout << "enter real and img part" << '\n';
cin >> real >> img;
}

 void output()
 {
 if(img<0)
 cout << real<< img << "i" << '\n';
else
 cout << real << "+"<< "i" << img << '\n';
}

 friend complex add(int,complex)
 friend complex add(complex,complex)
 };

 complex add(int a , complex s2)
 {
complex temp;
temp.real = s2.real + a;
temp.img = s2.img;
return temp;
 }

 complex add(complex s1, complex s2)
{
```

```
 complex s3;
s3.real = s1.real + s2.real;
s3.img = s1.img + s2.img;
return s3;
 }

void main()
{
 complex s1,s2,sum1,sum2;
int a;

clrscr();

s1.input();
s2.input();

cout << "first complex number is:"<< '\n';
s1.output();
cout << "second complex no is:"<< '\n';
 s2.output();
cout << "enter the value of a :";
 cin >> a;

sum1 = add(a,s2);
sum2 = add(s1,s2);

cout << "output of integer and complex no is:"<< '\n';
sum1.output();

cout << "output of complex and complex no is:"<< '\n';
sum2.output();

getch();
}
```

13.C++ Program on Constructors and Destructors

Program: Write a C++ program to invoke Constructor and Destructor.

```
#include<iostream.h>
#include<conio.h>
class byte
{
int bit;
int bytes;

public:

byte()
{
cout<<"\n Constructor invoked";
bit=64;
bytes=bit/8;
}

~byte()
{
cout<<"\n Destructor invoked";
cout<<"\n Bit = "<<bit;
cout<<"\n Byte = "<<bytes;
}
};

int main()
{
clrscr();
byte x;
byte();

x.byte :: ~byte();
return 0;
}
```

14.C++ Program on *this* pointer.

Program: Write a C++ program to use *this* pointer and return pointer reference.

```
#include<iostream.h>
#include<conio.h>
class number
{
int num;
public:

void input()
{
cout<<"\n Enter a number : ";
cin>>num;
}

void show()
{
cout<<"\n The minimum number :"<<num;
}

number min(number t)
{
if(t.num<num)
return t;
else
return *this;
}
};

void main()
{
clrscr();

number n,n1,n2;

n1.input();
n2.input();
n=n1.min(n2);
n.show();
}
```

15. C++ Program on *file I/O operations*.

Program: Write a C++ program to write text in the file. Read the text from the file from end of file. Display the contents of the file in reverse order.

```
#include<iostream.h>
#include<conio.h>

int main()
{
clrscr();

ofstream out;

char data[25];
out.open(“text”,ios::out);
cout<<”\n Enter text”<endl;
cin.getline(data,25);
out<<data;
out.close();

ifstream in;
in.open(“text”,ios::in);
cout<<endl<<”Reverse Contents of the file \n”;
in.seekg(0,ios::end);
int m=in.tellg();
char ch;

for(int i=1;i<=m;i++)
{
in.seekg(-i,ios::end);
in>>ch;
cout<<ch;
}
return 0;
}
```