Laboratory Manual for

EC404–Object Oriented Programming

B. Tech.

SEM. IV (EC)

Department of Electronics & Communication
Faculty of Technology
Dharmsinh Desai University
Nadiad

TABLE OF CONTENTS

PART I

LAB MANUAL

EXPERIMENT – 1

REVISITING C LANGUAGE

OBJECTIVE:
To gain experience with
1. Integer and floating-point numbers.
2. Writing arithmetic expressions in C.
3. Understand nested branches and loops.
4. Programming loops with for and do/while statements.

Writing simple programs that read numbers and text, process the input and display the results.

SAMPLE PROGRAM:

```
#include<stdio.h>
int main( )
{
        int odd(int);
        int i,k,a[10];
        for(i=0;i<10;i++)
        {
                printf("Enter a[%d]",i);
                scanf("%d",&a[i]);
        }
        for(i=0;i<10;i++)
        printf("a%d =  %d"i,a[i]);
        k=0;
        for(i=0;i<10;i++)
        {
              if(!odd(a[i]))
             {
                        k++;
                        printf("\n%d %d",a[i],k);
             }
        }
        printf("%d",k);

 }

int odd(int j)
{
                if(j%2)
                {
                        return (1);
                }
                else
                {
                        return (0);
}               }
```

EXERCISE:

(1) Write a program to find the factors of any number. Also find the prime factors of that number.
(2) Write a program to input a string and display it in reverse order.
(3) Bubbles sort an array of 10 integers. Arrange them in descending order.
(4) Define a structure *book_shop*. It should contain the name of the book, name of the author, number of copies available, price of the book and a Book ID No. Create a list of 5 such books. Write a function to search for a book using the name of the book. Ask the user for the number of copies he wants. Calculate the cost. If sufficient copies are not available, display appropriate message. Write a function to update the database after each purchase.
(5) Write a function to exchange the values of two variables. Use pass by reference method to exchange the data.

EXPERIMENT – 2

CLASSES AND OBJECTS

SAMPLE PROGRAM:

Define a class **Cube** that should contain its dimensions and the appropriate functions to get and display data. The class should include a function to calculate the volume of the cube using the cube dimensions.

```
#include "iostream"

using namespace std;

class Cube {

private:
        float height;
        float width;
        float length;

public:
        void SetData(float h, float w, float l)   {
                height = h;
                width = w;
                length = l;
        }

        void GetData( ) {
                cout << "Enter Height: ";
                cin >> height;
                cout << "Enter Width: ";
                cin >> width;
                cout << "Enter Length: ";
                cin >> length;
        }

        void ShowData( ) {
                cout << "Dimensions of the objects are:\n";
                cout << "Height: "<<height<<",Width: "<<width<<", Length: "<<length ;
        }

        void Volume( ) {
                cout << "\nVolume of the cube is: " << height * width * length << endl;
        }
};
```

```
int main( ) {
        Cube c1;
        c1.SetData(12.3, 23.4, 34.5);
        cout << "Object c1: " << endl;
        c1.ShowData( );
        //cout << c1.height;     //This statement will show error. Private data not accessible.
        c1.Volume( );

        Cube c2;
        cout << "\nEnter the content of object c2: " << endl;
        c2.GetData( );
        cout << "Object c2: " << endl;
        c2.ShowData( );
        c2.Volume( );

        return 0;
}
```

MODIFICATION:

1. Modify the above program to write all the function definitions outside the class.
2. Add a function, which displays a menu having the options to find the *surface area* and the *volume* of the cube. Write another function to calculate the *surface area* of the cube from the input dimensions. The menu itself should call the appropriate function to calculate *volume* or *surface area*.

EXERCISE:

1. Imagine an experiment in which you have to observe the performance of charging of a capacitor. You have to give the charging voltage, the value of the capacitance and resistance as inputs. The output should be a table showing the voltage across the capacitor with respect to time. Remember that the capacitor will be completely charged at time equal to the 5 times of time constant **τ**. So time should be varied from 0 till 5**τ** in suitable steps. Design a suitable class Experiment having appropriate functions to input the data, calculate the voltage and display the output.

EXPERIMENT – 3

(A) CONSTRUCTOR

OBJECTIVE:

To study the initialization of data members through constructors.

SAMPLE PROGRAM*:*

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

class Distance
{
   private:
      int mts;
      int cms;
   public:
            Distance ( ) {
                                    mts =0; cms=0;
                                }
            void GetData( );
            void Display( );
            void CalDist(Distance, Distance);
};

void Distance::GetData( )
{
                  cout << "\nEnter Distance in mts: ";
                 cin >>mts;
                cout << "Enter Distance in cms: ";
                cin >> cms;
}

void Distance::Display( )
{
       while((cms >=100) | (cms > 0 & mts < 0)) {
                    mts++;
                    cms -=100;
                                }
       while(cms < 0 & mts > 0) {
                    mts--;
                    cms +=100;
                                        }

            cout <<"Meters: "<<mts<<"\t Centimeters: "<<cms<<endl;
}
```

```
void Distance::CalDist(Distance d1, Distance d2)
{
        mts = d1.mts - d2.mts;
        cms = d1.cms - d2.cms;
}

int main()
  {
        Distance dist1, dist2,dist3;

        dist1.GetData( );
        dist2.GetData( );
        dist3.CalDist(dist1,dist2);
        cout << "Content of dist1 = "<<endl;
        dist1.Display( );
        cout << "Content of dist2 = "<<endl;
        dist2.Display( );
        cout << "Content of dist3 = "<<endl;
        dist3.Display( );
         return 0;
   }
```

MODIFICATION:

(1) Modify the above program to overload the Constructor to initialize new objects by passing parameters in two ways :

   a) Distance in mts and cms is passed.

   b) Only distance in mts is passed.

(2) Remove GetData( ) if it has become redundant.

(3) Modify each Constructor and Destructor such that it also display the number of times it is called.

(B) COPY CONSTRUCTOR AND DESTRUCTOR

OBJECTIVE:

To study the need of copy constructor and copy of data members through copy constructors.

SAMPLE PROGRAM:

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

class array
{
        int *p;
        int size;

public:
        array(int sz) {
                p = new int[sz];
                cout << "constructor created array at " << p << endl;
                cout << "pointer at " << &p <<endl;
                size = sz;
                      }

        //~array() { delete [] p; }      //Line -1

        //copy constructor

        /*array(const array &a) {       //Line - 2
                int i;
                p = new int[a.size];
                cout << "Copy constructor created array at " << p << endl;
                cout << "pointer at " << &p <<endl;
                for(i=0; i<a.size; i++) p[i] = a.p[i];
                                  }*/

        void put(int i) {
                if(i>=0 && i<size)
                     cin>>p[i];
                        }

        int get(int i) {
                return p[i];
                     }
        int* getp() {
                return p;
        }

};



int main()
{
        array num(10);
        cout << "New object at " << &num << endl;
        int i;
        for(i=0; i<10; i++) num.put(i);
        for(i=0; i<10; i++) cout << num.get(i);
        cout << "\n";
        // create another array and initialize with num
        array x(num); // invokes copy constructor
        cout << "Another object at " << &x << " and array at " << x.getp() << endl;
        for(i=0; i<10; i++) cout << x.get(i);
        cout << "\n";
        return 0;
}
```

MODIFICATION:

(1) Uncomment Line -1 and observe the output.

(2) Uncomment Line -1 and Line -2. Observe the output.

(3) Implement Sort( ) to sort an array in descending order and also display the sorted output. Use copy constructor in function call and function return statement.

(4) Modify each Constructor and copy constructor such that it also display the number of times it is called.

EXERCISE:

(1) Create a class Patient which has the following content: Name of Patient, Age, Sex, Ward Number, and Bed Number.Each ward has 2 beds, both of which should be filled before moving on to the next ward. Ward number and Bed number should be automatically initialized using the default constructor for each new patient. Use another constructor to initialize them using the explicit values passed as arguments.

Write a member function to do following:
a) To input details of patient
b) To modify a patient’s detail
c) To display patient’s detail

Write a program which creates an array of 5 patients. Also creates the menu which asks the user to choose from following tasks:

Enter the patient detail

Change the patient detail

Display the patient detail

Sort the array of patients according to their ages.

EXPERIMENT – 4

FUNCTIONS OVERLOADING

SAMPLE PROGRAM:

A sample program finds the volume of different shapes by using the concept of function overloading.

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

class Shape
{
        public:
        int volume (int);                      //To find volume of sphere
        double volume (double, int);           //To find volume of cylinder
        long volume (long, int, int);          //To find volume of cube
};
int Shape :: volume(int s)
{
        return (s*s*s);
}
double Shape :: volume(double r,int h)
{
        return(3.14159*r*r*h);
}
long Shape :: volume(long l,int b,int h)
{
        return(l*b*h);
}

void main( )
{
        Shape sphere, cylinder, cube;
        cout<<"VOLUME OF SPHERE   => "<<sphere.volume(10)<<endl;
        cout<<"VOLUME OF CYLINDER => "<<cylinder.volume(2.5,8)<<endl;
        cout<<"VOLUME OF CUBE     => "<<cube.volume(100L,75,15);
}
```

MODIFICATION:

1. Modify the above program to include a function Area which can calculate the surface areas of a sphere, a cylinder or a cube using the concept of function overloading.

EXERCISE:

1. Create a class Telephone Directory which has following information: Telephone number, Name of subscriber and Address. Write a member function to perform the following tasks:
    1. To input the details of subscriber
    2. To print the details of subscriber according to inquiry generated from user. Inquiry is either based on telephone number or name of subscriber.
    3. To modify the details of a particular subscriber, Modification is based on name of subscriber, if more than one match is found; it has to ask another field. Based on second field, it will modify the details.

    - Write a program for 10 subscribers,   which will creates the menu for following tasks:

        a) Enter the details of subscriber
        b) Inquiry for subscriber
        c) Modify the detail of subscriber

EXPERIMENT – 5

OPERATOR OVERLOADING

SAMPLE PROGRAM:

A sample program finds the volume of different shapes by using the concept of function overloading.

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

using namespace std;

class String {
char Name[25];
public: String(char *name) {
strcpy(Name,name);
}
int operator == (String);
};

int String::operator == (String S1) {
if(strcmp(Name,S1.Name))
return 0;
else
return 1;
}

int main ( ) {
String Str1("There");
String Str2("There");
if(Str1 == Str2)
cout<<"Both strings are equal"<<endl;
else
cout<<"Strings are not equal"<<endl;
return 0;
}
```

MODIFICATION:

1. Modify the above program to overload + operator to concatenate the two strings using operator overloading.

EXERCISE:

1. Use the class definition described in lab3 for the *Distance* class. Overload -- operator to perform following operation in which it subtracts 1 from meters member and check that distance should not be negative.

   dist1--;

   Function should allow to use the result in other operation like *dist2* = *dist1--;* and also write modified function which differentiate postfix and prefix notation.

EXPERIMENT – 6

TYPE CONVERSION

SAMPLE PROGRAM:

Following program demonstrates how to convert basic types to objects and vice versa. Here class *Time* encapsulates time of one particular time zone.

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

class Time
{
	int hrs, mins, secs;
public:
	Time( )
	{
		hrs = mins = secs = 0;
		cout <<  “Default Constructor called.”;
	}
	Time(int h, int m, int s)
	{
		hrs = h; mins = m; secs = s;
		cout <<  “Three argument Constructor called.”;
	}
	Time(int s)	//One argument constructor to convert from basic to user defined type.
	{
		hrs = s / 3600;
		s %= 3600;
		mins = s / 60;
		secs = s % 60;
		cout <<  “One argument Constructor called.”;
	}
	operator int( )  //Conversion function to convert from user defined to basic type.
	{
		int s = (hrs * 3600) + (mins * 60) + secs;
		return s;
	}
	void getTime();
	void showTime();
};
void Time :: getTime()
{
	cout<<"Enter hours: ";cin >> hrs;
	cout<<”Enter minutes: ”;	cin >> mins;
	cout<<”Enter seconds: ”;	cin >> secs;
}
void Time :: showTime( )
```

```
{
        cout << hrs <<  " : "  << mins << " : "  << secs << endl;
}

void main( ){

        Time t1 = 3800;
        cout << "Time t1: ";
        t1.showTime( );
        Time t2(10, 20, 40);
        cout << "Time t2: ";
        t2.showTime( );
        int s = t2;                             //Implicit casting
        // int s = int(t2);                     //Explicit casting
        cout << "Time t2 in seconds:" << s;
}
```

EXERCISE:

1. Create two classes *Degree* and *Radian* to store two angles in degree and in radian respectively. Assign object of one class with other. Do appropriate conversion for this assignment. Declare, perform operation and show the results in main. Show both possible way of conversion.

EXPERIMENT – 7

INHERITANCE

SAMPLE PROGRAM:

```
#include <iostream>
using namsespace std;

class X {
protected:
int a;
public:
X(void);
};

class Z : public X {
public:
Z(void);
int make_aa(void);
};

X::X(void) { a=10; cout<<"initializing X\n"; }
Z::Z(void) { cout<<"initializing Z\n"; }
int Z::make_aa(void){ return a*a; }

main(void){
Z i;
cout<<i.make_aa( );
return 0;
}
```

MODIFICATIONS:

1. Modify the above program to include one more data member int b and write a member function for multiplication of a and b in class Z.
2. Modify the above modification program to include a constructor in class Z to initialize a and b.

EXERCISE:

1. Design a class Student with data members: Name, ID and Semester. Derive another class Sessional with an array of 5 integers to store sessional marks of 5 subjects. Derive a class External with an array of 5 integers to store the marks of 5 subjects. Write suitable member functions to initialize the data members of each class and produce the marksheet of the student containing the marks of sessional and external examination with total marks as well.

2. What should be the total memory requirement for the object of class Z and External? How can it be verified in the program?
3. Will the sample program work correctly, if the access specifier is made private with X while deriving Z? Explain your answer.

EXPERIMENT – 8

POLYMORPHISM

SAMPLE PROGRAM:

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

class convert {
protected:
        double val1;  // initial value
        double val2;  // converted value
public:
        convert(double i) {
                        val1 = i; }
        double getconv( ) { return val2; }
        double getinit( ) { return val1; }
        virtual void compute( ) = 0;
};

 // Liters to gallons.
class l_to_g : public convert {
public:
       l_to_g(double i) : convert(i) { }
       void compute( ) {
            val2 = val1 / 3.7854; }
};

// Fahrenheit to Celsius
class f_to_c : public convert {
public:
      f_to_c(double i) : convert(i) {  }
      void compute( ) {
         val2 = (val1-32) / 1.8; }
};

int main( ) {
      convert *p;  // pointer to base class
      l_to_g lgob(4);
      f_to_c fcob(70);

       // use virtual function mechanism to convert
      p = &lgob;
      cout << p->getinit( ) << " liters is "; p-
      >compute( );
      cout << p->getconv( ) << " gallons\n";  // l_to_g
```

```
    p = &fcob;
    cout << p->getinit( ) << " in Fahrenheit is "; p-
    >compute( );
    cout << p->getconv( ) << " Celsius\n";  // f_to_c

    return 0;
}
```

MODIFICATION:

1. Modify the above program to include the class *f_to_m* which will convert the feet in to meters.

EXERCISE:

1. Create a class called *Number* has one data member of type integer. There two member function *Setvalue( )* which will set the value of data member and *show( )* to display the value. Drive a class *HexType*, *DecimalType* and *OctType* from base class which redefined the *show()* will display the value of data member of base class in respective number base.

EXPERIMENT – 9

FILE MANAGEMENT IN C++

SAMPLE PROGRAM:

```
#include <iostream>
#include <fstream>
#include <cstring>
#include<iomanip>

using namespace std;

class Inventory  {
          char name[20];
          int code;
          float cost;
public:
          Inventory ( ) { };
          Inventory(char *n, int cd,float ct){
                    strcpy(name, n);
                    code = cd;
                    cost = ct;
          }

void getdata( ) {
          cout<<"Enter name:";
          cin>>name;
          cout<<"Enter code:";
          cin>>code;
          cout<<"Enter cost:";
          cin>>cost;
}

void putdata( ) {
          cout<<setw(10)<<name<<endl;
          cout<<setw(10)<<code<<endl;
          cout<<setw(10)<<cost<<endl;
}
};

int main( ) {
          Inventory item,item1;
          char c;
          fstream infile("file.txt", ios::in | ios::out | ios::binary);
          do {
                              cout << "1. Enter numbers\n";

cout << "2. Display numbers\n";
                              cout << "3. Quit\n";
```

```
                cout << "\nEnter a choice: ";
                cin >> c;


        switch(c) {
            case '1':
                item.getdata( );
                infile.write((char *)&item,sizeof(item));
                break;
            case '2':
                infile.seekg(0,ios::beg);
                while(infile.read((char *)&item1,sizeof(item1)))
                {
                item1.putdata( );
                }
                cout << endl;
                break;
            case '3':
                infile.close( );

        }
        } while(c !='3');
        return 0;
}
```

MODIFICATION:

1. Modify the above program that modifies the detail of items that is already present in file and also add new item information in file.

EXERCISE:

1. Write a program to split the content of file that is created by sample program into two files. One file has name of all items present in stock.txt and Second file has cost of all item present in stock.txt using command line argument.

EXPERIMENT – 10

EXCEPTION HANDLING

SAMPLE PROGRAM:

```
#include<iostream>
using namespace std;
class Dummy{
        private:
        int n;
        public:
        Dummy(){}
        Dummy(int n){
        n=m;
        }
        int divide(int a){
                if(a==0)
                        throw a;
                cout<<"\n returning from divide"<<endl;
        return (n/a);
        }
};

int main( ){
        Dummy d1;
        int b,c;
        bool flag;
        d1=5;
        flag=true;
        while(flag){
                cout<<"\n enter b\n";
                cin>>b;
                try{
                        c=d1.divide(b);
                        flag=false;
                }
                catch(int x){
                        cout<<"attempt of division by zero\n";
                        cout<<"check your input again\n";
                }
}
        cout<<"\nans="<<c<<endl;
        return 0;
}
```

MODIFICATION:

1. Understand the above program thoroughly and modify it in such a way that the user is given maximum three attempts of division to re execute the code after an attempt of division by zero. The program should exit otherwise.

Hint: Use rethrowing of exception.

EXERCISE:

1. Modify the above modified program further so as to display a suitable warning message before exiting the main due to an exception that is rethrown.

Hint: Use your own terminate ( )

EXPERIMENT – 11

TEMPLATES

SAMPLE PROGRAM:

```
#include<iostream>
using namespace std;
template <class T>
void swap1 (T & p, T& q)
{
        T temp;
        temp = p;
        p = q;
        q =temp;
}
int main( )
{
        int a,b;
        float x,y;

        cout<<"Enter two integers:";
        cin>>a>>b;
        cout<<"Enter two floats:";
        cin>>x>>y;

        swap1(a,b);
        swap1(x,y);

        cout<<"Value of integers after swapping:";
        cout<<a<<"\t"<<b;
        cout<<endl;
        cout<<"Value of floats after swapping:";
        cout<<x<<"\t"<<y;
        cout<<endl;

        return 0;
}
```

MODIFICATION:

1. Modify the above program to demonstrate the use of explicit specialization of a template function *swap1( ).*

EXERCISE:

1. Define a template class *Array* that will have three member functions given below:
   a) *sort( )* that will sort the elements in ascending order.
   b) *reverse( )* that will reverse the content of an array and
   c) *safe( )* that will check the size of array; if size is exceeded terminate the program otherwise perform the above two task.

2. Rewrite the above program in such a way that it has one non-type argument in template function as well as default arguments to template function.

PART II

TUTORIAL

TABLE OF CONTENTS

DHARMSINH DESAI UNIVERSITY, NADIAD

(Faculty of Technology)

B. Tech. Sem-IV (EC), Subject: Object Oriented Programming

ASSIGNMENT 1

1) The binding of data and functions together into a single class-type variable is referred to as

a) Data Encapsulation

b) Data Abstraction

c) Data hiding

d) All of above

2) Which of the following statement is correct?

a) Overloaded functions can accept same number of arguments.

b) Overloaded functions always return value of same data type.

c) Overloaded functions can accept only same number and same type of arguments.

d) Overloaded functions can accept only different number and different type of arguments.

3) Pick out the correct statement related friend function.

a) friend function can be a member of another class.

b) friend function can not be a member of another class.

c) friend function can or can not be a member of another class.

d) None of the above

4) Inline function

a)cannot used for recursive function b)cannot used for static variables
c)cannot used for large function definition d)All of above

5) What is the value of x and how many copy of x is generated after executing following code?

```
#include<iostream>
using namespace std;
class Data
{
public:
static int x;
Data() { x=10;}
int increment( ) { x++; return x;}
};
int Data::x;
int main()
{
Data D1,D2;
D1.increment();
D2.increment();
cout<<Data::x;
return 0;
}
```

a) 11 ,1 b)11,2 c) 12,1 12,2

6) Which of the following is correct for constant pointer?

a) int x=10, y=20 , * const ptr= &x; ptr=&y;

b) int x=10,y=20 , *const ptr = &x; *ptr =30;

c) int x=10, y=20 , const *ptr= &x; ptr=&y;

b) int x=10,y=20 , const *ptr = &x; *ptr =30;

7) A constructor requires at least

a) A statement to initialize one of the data members. c) *return* statement.

b) Statements to initialize all data members. d) Nothing.

8) Which of the following correctly declares a reference for int myCreditHours;?

a) int &cr = myCreditHours; c) int *cr = myCreditHours;
b) int cr = &myCreditHours; d) int *cr = &myCreditHours;

9) Destructors are invoked___________.

a) explicitly when needed c) explicitly when object goes out of scope
automatically when object goes out of scope d) automatically at the end of a program

10) A constant member function can modify

a) private data member c) mutable private data member

b) public data member d) none

DHARMSINH DESAI UNIVERSITY, NADIAD

(Faculty of Technology)

B. Tech. Sem-IV (EC), Subject: Object Oriented Programming

ASSIGNMENT 2

1. How can a data member be modified even by a const function?
(a) It is never possible.

(b) It is possible without any specification.

(c) It is possible when the member is declared mutable.

(d) It is possible when the member is declared const.

2. What can be written in the blank with the function definition?
<return type><function name>(argument1, ...)

{

<function body>

}

(a) static (b) const (c) volatile (d) friend

3. When overloading unary operators___________.
(a) no argument is passed explicitly

(b) one argument is to be passed explicitly

(c) no argument is passed implicitly

(d) no argument is passed explicitly only if overloaded as member functions

4. The operators created using friend function can also be created by member functions.

(a) Always true (b) Always false (c) Partially true (d) Never true

5. The advantage of operator overloading is___________.
(a) better readability (b) easy usage (c) easy coding (d) Both (a) and (b)

6. Using operator overloading, one can___________.
(a) design new operators

(b) not design new operators

(c) only overload available operators

(d) give a meaning to all the available operators

7. Which of the following is the correct syntax of overloading the new operator for a class named Test?
(a) void Test operator new(int size);

(b) void * Test operator new(int size);

(c) void * Test operator new(size_t size);

(d) Test * Test operator new(size_t size);

8. When a member is protected, it is not different from___________unless the class is inherited.
(a) public (b) private (c) Both (d) None

9. The redundancy reduction reduces___________.
(a) code size (b) chances of errors (c) Both (d) None

10. A copy constructor must be called
(a) When an object is initialized

(b) When a function returns a object of class

(c) When objects are passed as an argument of function

(d) All of above

DHARMSINH DESAI UNIVERSITY, NADIAD

(Faculty of Technology)

B. Tech. Sem-IV (EC), Subject: Object Oriented Programming

ASSIGNMENT 3

1. When a programmer does not want a class to be instantiated, it can be achieved by __________.
   (a) defining a member function inside that class

   (b) defining a friend function inside that class

   (c) defining a virtual function inside that class

   (d) defining a pure virtual function inside that class

2. The objects that can be stored and retrieved later are known as___________.
   (a) serial objects

   (b) serialized objects

   (c) persistent objects

   (d) persisted objects

3. Text file storage and retrieval may require___________.
   (a) character conversions

   (b) numeric conversions

   (c) Both

   (d) None

4. Which of the following is the correct syntax for declaring function templates?
   (a) template <typename T>; void fun(T var1);

   (b) template <T>; void fun(T var1);

   (c) template <type T>; void fun(T var1);

(d) template <tname T>; void fun(T var1);

5. What are the classes Stack <int> and Stack <char> known as?
   (a) Normal classes

   (b) Template classes

   (c) Specializations

   (d) Special classes

6. If we have object from ofstream class, then default mode of opening the file is______.
   (a) ios::in

   (b) ios::out

   (c) ios::in|ios::trunc

   (d) ios::out|ios::trunk

7. A try block throwing an exception can have
   (a) one catch block (c) any number of catch blocks

   (b) no catch block (d) catch blocks with argument of basic data type only

8. A throw statement used to rethrow an exception has
   (a) one argument of basic data type (c) any number of arguments

   (b) no argument (d) one argument of any data type

9. Predict the output of following program.

Assume that the size of char is 1 byte and size of int is 4 bytes, and there is no alignment done by the compiler.

```
#include<iostream>

#include<stdlib.h>

using namespace std;
```

```
template<class T, class U>
class A  {
    T x;
    U y;
    static int count;
};

int main()  {
   A<char, char> a;
   A<int, int> b;
   cout << sizeof(a) << endl;
   cout << sizeof(b) << endl;
   return 0;
}
```

(a) 6　　(b) 2　　(c) 8　　(d) Compiler error

12　　8　　8

10. In polymorphism, members are accessed using

(a) object name　　(c) pointer to the base class

(b) class name　　(d) pointer to the class of the member to be accessed

PART III

SESSIONAL QUESTION PAPER

DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B.TECH. SEMESTER IV [EC]
SUBJECT: (EC404) OBJECT ORIENTED PROGRAMMING

Examination	**: First Sessional**	**Seat No.**	**: ___________**
Date	**: 11/01/2019**	**Day**	**:Friday**
Time	**: 10 :30 to 11:45 am**	**Max. Marks**	**36**

INSTRUCTIONS:

1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 **(A) Do as directed.**

(I) State true/false with reason(s). **[3]**

(1) The constructor is called each time the object declaration is encountered.

(2) The constructor for a class with a pointer as data member needs to be appropriately defined.

(II) Choose the most appropriate answer(s). **[3]**

(1) If you attempt to create an object for which there is no constructor,
(a) run time error results. (b) a compile-time error results.
(c) compiler provides a constructor. (d) none.

(2) A class can have
(a) one object (b) ten objects (c) hundred objects (d) infinite objects

(3) A constructor requires at least
(a) a statement to initialize one of the data members.
(b) statements to initialize all data members.
(c) return statement.
(d) nothing.

(B) Do as directed.

(I) Choose the most appropriate answer(s). **[2]**

(1) If FPtr is a void pointer and SPtr is an integer pointer, which of the following statement(s) is valid in C++?
(a) FPtr = SPtr, (b) SPtr = FPtr, (c) both, (d) none

(2) Which of the following statement(s) is invalid for default argument?
(a) Pass(int Sub1, int Sub2)
(b) Pass(int Sub1, int Sub2, int Marks1 = 50)
(c) Pass(int Sub1, int Sub2, int Marks1 = 50, int Marks2 = 35)
(d) Pass(int Sub1, int Marks1 = 50, int Sub2, int Marks2 = 35)

(II) Write the output of given program. **[2]**

```
  #include <iostream>
using namespace std;
int main( )
{
int OriginalVariable;
int& ReferenceVariable = OriginalVariable;
OriginalVariable = 100;
ReferenceVariable = 200;
OriginalVariable++;
cout << OriginalVariable <<"\n";
ReferenceVariable++;
cout << ReferenceVariable <<"\n";
}
```

(III) Define the context switching and explain its significance. **[2]**

Q. 2. Attempt ***Any Two*** from the following questions. **[12]**

(A) (I). What is the difference between constant pointer and pointer to constant? Also write their declarations. **[3]**

(II). What is the value of *x* and how many copies of *x*are generated after executing the following code? **[3]**

```
#include <iostream>
using namespace std;
class Data
{
public:
static int x;
Data() { x=10;}
int increment( ) { x++; return x;}
};
```

```
int Data::x;
int main( )
{
Data D1,D2;
D1.increment();
D2.increment();
cout<<Data::x;
return 0;
}
```

(B) Write an object oriented program to swap and print two integer numbers. Define function SwapInt( ), which passes two reference variables and has reference return value. **[6]**

(C) Write an object oriented program to calculate and print area of the rectangle. Define Area as a class with two member functions Getdata( ) and Printdata( ) to calculate and print the area respectively. Take the array of three objects. **[6]**

Q.3 **(A)** Predict the output of following program and answer the questions. **[6]**

(I). Which constructor will be called in main? When? How many times? Why?

(II). What will be the effect on program execution, if the constructor with argument is defined as private?

```
#include "iostream"
using namespace std;
class Item {
        int ItemNo;
public: Item( ) {   }
        Item (int j) {
        ItemNo = j;
}
void SetNo(int TempNo) {
        ItemNo = TempNo;
        }
        void ShowDetails() {
        cout<<"\nItem"<< ItemNo;
        }
```

```
int main() {
        Item Array[12];
        for(int i=0; i<12; i++)
                Array[i].SetNo(i);
        for(int i=0; i<12; i++)
                Array[i].ShowDetails();
        return 0;
}
```

```
};
```

(B) Define an examiner class only. It contains the number of answer sheets to be examined, number of subjects, college name, type of examiner (1 for Internal or 2 for External) . Write an appropriate overloaded member function Calculate ( ) to calculate remuneration given to examiner(if it is Internal, Rs. 10 / paper and for External Rs. 20 / paper). Member function Calculate ( ) should be performed using (i). type of examiner, (ii) college name. **[6]**

OR

Q.3. (A) Identify and correct errors in the class and the *main( )* given below. Also predict the corresponding output. **[6]**

```
#include "iostream"
using namespace std;
class Counter {
        private:
        unsigned int count;
        void countplus( ) {
                        count++;
                        }
Counter(int c) {
                        count = c;
                        }
public:
        unsigned int getcount( ) {

                        }
};
```

```
int main( ) {
        Counter c1;
c1 = 0;
        Counter c2 = Counter(5);
        cout <<"\nCounter 1 = "<< c1.count;
        cout <<"\nCounter 2 = "<<
c2.getcount( );
        countplus( );
        cout <<"\nCounter 2 = "<<
c2.getcount( );
        return 0;
}
```

(B) Define an appropriate class definition for class called Person for following main( ). Is it possible to make Elder ( ) as a friend function? If yes, write appropriate changes? **[6]**

```
#include "iostream"
using namespace std;
int main( )
{
Person P1,P2,P3;
P1.SetDetails("Name", 5.6,32); // Name, Height, Age
P2.SetDetails("Name1", 5.8,38);
P3=P1.Elder( P2); // Function finds elder person
P3.Display( );
return 0;
}
```

DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B.TECH. SEMESTER IV [EC]
SUBJECT: (EC404) OBJECT ORIENTED PROGRAMMING

Examination	**: Second Sessional**	**Seat No.**	**: ___________**
Date	**: 15/02/2019**	**Day**	**:Friday**
Time	**: 10:30 to 11:45**	**Max. Marks**	**36**

INSTRUCTIONS:
1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q. 1. (A) Do as directed.

(I) State true/false with reason(s). **[3]**

(1) A programmer must provide copy constructor in the class definition.

(2) Use of member initialization list makes the program more efficient.

(II) Choose the most appropriate answer(s). **[2]**

(1) What will support the program to execute the following statement successfully?
Aclass A = 10; /* Aclass is class name. */
(a) a default copy constructor. (b) a user defined copy constructor.
(c) one argument constructor. (d) default constructor.

(2) For which of the followings, use of member initialization list is compulsory?
(a) integers. (b) strings. (c) characters. (d) constants.

(II) What is the major benefit of using constructor with default arguments? **[1]**

(B) Do as directed.

(I) Choose the most appropriate answer(s). **[2]**

(1) Which of the following is the correct syntax of overloading the new operator for a class named Test?
(a) void Test operator new(int size); (b) void * Test operator new(int size);
(c) void * Test operator new(size_t size); (d) Test * Test operator new(size_t size);

(2) What will the following statement do?
Matrix operator *(int multiplier, Matrix tempMatrix);
(a) Overload the * (multiplication) operator for Matrix class.
(b) Compiler will generate an error.
(c) Overload the * (pointer) operator for Matrix class.
(d) inker will generate an error .

(II) Identify and correct the syntax and logical error(s) in following code to get output as *5*. **[2]**

```
#include "iostream"
using namespace std;
int main()
{
int RollNumber;
void PrintDetails() const
student OtherStudent;
```

```
        OtherStudent.PrintDetails();
        OtherStudent.RollNumber = 5;
    cout << OtherStudent.RollNumber;
    }
```

(III) For the following program, identify the error(s) and resolve them, if any. **[2]**

```
class X {
protected:
        int  x;
public:
X( ) { x= 2;}
};
class Y:public X {
protected:
        int  y;
public:
        Y( ) { y = x+1;}
};
```

```
class Z:public X {
protected:
        int  z;
public:
        Z ( ) { z= x+2;}
};
class W:public Y, public Z {
        int  w;
public :
        W( ) { w = x+y+z;}
};
main( ) {
W  ww; return 0;
}
```

Q. 2. Attempt ***Any Two*** from the following questions. **[12]**

(1) (I). Write appropriate class definition and member functions for given main below. (C1 is object of class Cartesian, P1 is object of class Polar.) **[4]**

```
int main( ){
Cartesian C1;
Polar P1(10,45);
    C1=P1;
    C1.show( );
    }
```

(II). What is the difference between private and protected access modifier (visibility mode) in inheritance? **[2]**

(2) Overload ++ in prefix and postfix version for the Complex class. Prefix ++ adds to the real part and postfix adds to the imaginary part. **[6]**

(3) Overload operator '/' such that division from and to normal integer work the same way. **[6]**

Q. 3. **[6]**

(1) Write a program with appropriate class definition to show unavoidable requirement of a copy constructor. Also discuss the problem(s) that can arise in absence of the copy constructor in the same program. **[6]**

(2) Identify the syntax and logical errors in following code. Predict the output after correction. What is the size of object *obj* of C class? Write an appropriate constructor in each class to initialize the data members.

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

```
int main() {
C obj;
```

```
class A {
int a;
public:
void seta( ) { a=5;}
};
class B : public A {
int b;
void setb( ){ b=a+2;}
};
class C: public B {
int c;
public:
void setc( ) { c=a+b;}
void show( ) {
cout<<"a="<<a<<"\nb="<<b<<"\nc="<<c<<endl;}
};
```

```
obj.seta();
obj.setb();
obj.setc();
obj.show();
return 0;
}
```

OR

Q. 3.

(1) Define a class called Vehicle. Inherit this class into TwoWheelers and FourWheelers. Provide a function that calculates the mileage of the vehicle. (Distance traveled in km/fuel consumed). **[6]**

(2) Predict the output of following code. **[6]**

```
#include <iostream>
using namespace std;
class CheckOrder {
public:
int Bottom;
int Top;
int Middle;
CheckOrder(int i)
:Middle(Bottom/10),
Top (i/10),
Bottom(i) { }
CheckOrder(unsigned i)
        :Middle (i/=10),
        Top (i/=10),
        Bottom (i/=10)
        {       }
void PrintI() {
cout<<"Top = "<<Top<<"\n";
cout<<"Middle = "<<Middle<<"\n";
cout<<"Bottom = "<<Bottom<<"\n";
        }
};
```

```
int main() {
        int si = 4567;
        unsigned ui = 4567;
        CheckOrder CO(si);
        CheckOrder CO2(ui);
        cout<<"\n"<<"Case 1"<<"\n";
        CO.PrintI();
        cout<<"\n"<<"Case 2"<<"\n";
        CO2.PrintI();
        return 0;
}
```

DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B. TECH. SEMESTER IV [ELECTRONICS & COMMUNICATION]
SUBJECT: (EC404) OBJECT ORIENTED PROGRAMMING

Examination	**: Third Sessional**	**Seat No.**	**: ___________**
Date	**: 02/04/2019**	**Day**	**:Tuesday**
Time	**: 10:30 – 11:45**	**Max. Marks**	**: 36**

Q-1 (A) Choose the most appropriate(s). **[04]**

(1) The function unexpected( ) will be called in case the exception is
- (a) not included in the list of restricted exceptions.
- (b) from a destructor.
- (c) rethrown without any active exception.
- (d) all of above.

(2) Which of the followings can definitely handle an exception?
- (a) provision of try and catch blocks in the program.
- (b) Exception must be thrown from try block and catch block following it.
- (c) Exception is thrown from try block and a catch(...) block following it.
- (d) Provision of a throw statement in the program.

(3) When providing a destructor in the class definition is advisable?
- (a) When a pointer is used in the program.
- (b) When a pointer is used in the class.
- (c) When memory is dynamically allocated while creating an object.
- (d) All of above.

(4) In polymorphism, members are accessed using
(a) object name (c) pointer to the base class
(b) class name (d) pointer to the class of the member to be accessed.

(B) Do as directed. **[08]**

(1) When the base class pointer points to derived class object, which function is called for following situations? [2]
(i)if base class and derived class having same function show( ).
(ii) if base class and derived class having same function show( ) but base class has virtual show function.

(2) Write syntax of a) Partial Specialization b) Explicit Specialization. [2]

(3) Differentiate text file and binary file. [2]

(4) How many times Hello message will be printed for following code? If test file contains DDUniversity. [2]

```
#include<iostrem>
using namespace std;
int main( )
{ ifstream in(“test”, ios::in | ios::binary);
while(!(in.eof())) {
in.get(ch);
cout << “Hello\n”;
retrun 0; }
```

Q.2 Answer the following questions. (Attempt any three) **[12]**

(1) Write a template function definition only which will search a particular element is present in data or not. If it is present return non-zero value, otherwise zero. Is it required to use non-generic type in this template definition? If yes give syntax of it. [4]

(2) Answer the following question with reference to below mentioned class definitions. [4]
I) Is it possible to convert the base class in to an abstract class? If yes, covert base class into abstract class and make necessary changes in class definition to display title and price also.

```
class Media {
protected:
        char title[50];
        float price;
public :
Media(char  *s, float a){
strcpy(title,s);price =a; }
public :
Book(char *s, float a, int p) :Media(s,a)
{   pages =p;   }
void display( ){
cout<<”pages=”<<pages<<endl; }
};
class Disk:public Media {
```

```
void display( ) {
cout<<"title="<<title<<"\n
price="<<price<<endl;}
};
class Book:public Media {
        int  pages;
```

```
        float time;
public :
Disk(char *s, float a, float t):Media(s,a)
{  time =t;  }
void display( ){cout<<"time="<<time<<endl; }
};
```

(3) What is polymorphism? What are the types of polymorphism? What is the role of virtual keyword along with member function? [4]

(4) Write a Queue class as template class only. It contains two member function Push( ) and Pop( ) which will write into queue and read data from queue. [4]

Q.3 (1) Write a program to read the content of text file and split the content of this file into two files of equal size. [6]

(2) Answer followings for the program given below. [6]

(i) What type of exceptions are thrown?
(ii) Does the program handle all exceptions thrown? If yes, show how are they handled. If no, explain why they remain unhandled.
(iii) What changes are needed to ensure that no exception remains unhandled?

```
#include <iostream>
using namespace std;
class D {
public: ~D( ) {
                cout <<"Destructor Called\n";
                throw 1;
        }
};
int main( ) {
        D derived;
        try {
                throw derived;
        }
        catch(D d) {
        cout <<"This won't execute.\n";
}
return 0;
}
```

OR

Q.3 (1) Write a program to write details of 50 students in file and read the details of all students from the file and display on terminal. Use your own class definition for student class. [6]

(2) Answer followings for the program given below. [6]
(i) Is it necessary to write a destructor function following program?
(ii) Mention the order in which the objects will be destroyed? Explain in brief.
(ii) Write an appropriate destructor to verify that the objects are destroyed in the same order.

```
#include <iostream>
using namespace std;
class myclass {
public:
        int who;
myclass(int x) {who=x;}
} glob_ob1(1), glob_ob2(2);
int main() {
        myclass local_ob1(3);
        myclass local_ob2(4);
        return 0;
}
```