PROGRAMS

1. **Write a C++ Program to display Names, Roll No., and grades of 3 students who have appeared in the examination. Declare the class of name, Roll No. and grade. Create an array of class objects. Read and display the contents of the array.**

```
#include <iostream>
using namespace std;
#define MAX 10
class student
 {
    private:
       char name [30];
       int   rollNo;
       int   total;
       float perc;
     public:
   void getDetails(void);       //member function to get student's details
   void putDetails(void);       //member function to print student's details
  };
void student:: getDetails(void)  //member function definition, outside of the class
{
      cout << "Enter name: “;
     cin >> name;
     cout << "Enter roll number: ";
    cin >> rollNo;
    cout << "Enter total marks out of 500: ";
    cin >> total;
     perc=(float)total/500*100;
}
```

```
void student:: putDetails(void) //member function definition, outside of the class
 {
    cout << "Student details:\n";
 cout << "Name:"<< name << “, Roll Number:" << rollNo << “, Total:" << total << “, Percentage:"
       << perc;
 }

int main ()
   {
       student std[MAX] ;     //array of objects creation
       int n, loop;
        cout << "Enter total number of students: ";
        cin >> n;
        for (loop=0; loop< n; loop++)
          {
             cout << "Enter details of student " << loop+1 << ":\n";
             std[loop] .getDetails();
          }
 cout << endl;
 for(loop=0;loop< n; loop++)
  {
    cout << "Details of student " << (loop+1) << ":\n";
    std[loop]. putDetails();
  }
return 0;
}
```

Output:

Enter total number of students: 3

Enter details of student 1:

Enter name: ravi

Enter roll number: 123

Enter total marks out of 500: 234

Enter details of student 2:

Enter name: john

Enter roll number: 256

Enter total marks out of 500: 342

Enter details of student 3:

Enter name: seetal

Enter roll number: 76

Enter total marks out of 500: 455

Details of student 1:

Student details:

Name: ravi, Roll Number:123,Total:234,Percentage:46.8Details of student 2:

Student details:

Name: john, Roll Number:256, Total:342,Percentage:68.4Details of student 3:

Student details:

Name: seetal, Roll Number:76, Total:455, Percentage:91

2. Write a C++ program to create a class triangle, read the length and height, calculate the area and display the result.

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

  class Triangle
{
      private:
          float length;
          float height;
       public:
                  // Method to set the length and height
       void setDimensions(float l, float h)
       {
           length = l;
           height = h;
        }
 // Method to calculate the area of the triangle
     float area()
    {
         return 0.5 * length * height;
     }
 // Method to display the area
 void displayArea()
   {
    cout << "The area of the triangle is: " << area() << endl;
   }
};
```

```
 int main ()
{
   Triangle t;
   float l, h;
   cout << "Enter the length of the triangle: ";
   cin >> l;
   cout << "Enter the height of the triangle: ";
   cin >> h;
   t.setDimensions(l, h);
   t.displayArea();
   return 0;
}
```

Output:

```
Enter the length of the triangle: 4
Enter the height of the triangle: 6
The area of the triangle is: 12

Enter the length of the triangle: 7
Enter the height of the triangle: 8
The area of the triangle is: 28
```

3. Write a C++ program to find the area of rectangle. Use constructors to initialize length and breadth.

```
#include <iostream>
using namespace std;
class Rectangle {
   private:
      float length;
      float breadth;

   public:
                                        // Constructor to initialize length and breadth
      Rectangle (float l, float b)
        {
          length = l;
          breadth = b;
      }

                                    // Method to calculate the area of the rectangle
      float area (   )
        {
          return length * breadth;
        }

                                    // Method to display the area

      void displayArea ( )
        {
         cout << "The area of the rectangle is: " << area () << endl;
        }
    };

int main()
  {
   float l, b;
   cout << "Enter the length of the rectangle: ";
   cin >> l;
   cout << "Enter the breadth of the rectangle: ";
   cin >> b;
   // Create rectangle object using constructor                Output:

   Rectangle rect (l, b);                                  Enter the length of the rectangle: 2
      rect. displayArea ( )                                Enter the breadth of the rectangle: 3
      return 0;                                              The area of the rectangle is: 6
   }
```

4. Write a C++ program to overload binary + operator.

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

class complex {
   int a, b;
public:

   void getvalue () {
      cout << "Enter the value of Complex Numbers a,b:";
      cin >> a>>b;
   }

   complex operator+ (complex ob) {
      complex t;
      t.a = a + ob. a;
      t.b = b + ob. b;
      return (t);
   }
   void display () {
      cout << a << "+" << "i" << b << "\n";
   }

    };

int main ()
 {
   complex obj1, obj2, result, result1;

   obj1.getvalue();
  // obj2.getvalue();

   result = obj1 + obj2;

   cout << "complex no is:\n";
   obj1.display();

    }
```

Output:

```
Enter the value of Complex Numbers a, b:2 3

complex no is:

2+i3
```

5. Write a C++ program to create multilevel inheritance.

```
#include <iostream>
using namespace std;
class base //single base class
{
        public:
        int x;
        void getdata()
        {
        cout << "Enter value of x= "; cin >> x;
        }
};
class derive1 : public base // derived class from base class
{
        public:
        int y;
        void readdata()
        {
           cout << "\nEnter value of y= "; cin >> y;
        }
};
class derive2 : public derive1   // derived from class derive1
{
        private:
        int z;
        public:
        void indata()
        {
        cout << "\nEnter value of z= "; cin >> z;
        }
```

```
        void product ()
        {
           cout << "\nProduct= " << x * y * z;
        }
};
int main ()
{
    derive2 a;    //object of derived class
    a.getdata();
    a.readdata();
    a.indata();
    a.product();
    return 0;
}
```

Output:

```
Enter value of x= 3

Enter value of y= 4

Enter value of z= 6

Product= 72
```

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

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

class Weapon
    {
       public:
      virtual void features()
            {
               cout << "Loading weapon features.\n";
            }
     };
      class Bomb : public Weapon
            {
              public:
                  void features ()
                        {
                          this->Weapon::features ();
                        cout << "Loading bomb features.\n";
                            }
            };
 class Gun : public Weapon

          {
               public:
                void features()
                 {
                  this->Weapon::features();
                  cout << "Loading gun features.\n";
```

```
                    }
                };
            class Loader
                {
                    public:
                    void loadFeatures(Weapon *weapon)
                        {
                            weapon->features();
                        }
                };
  int main ()
  {
  Loader *l = new Loader;
  Weapon *w;
  Bomb b;
  Gun g;
  w = &b;
  l->loadFeatures(w);
  w = &g;
  l->loadFeatures(w);
  return 0;
}
```

Output:

```
Loading weapon features.
Loading bomb features.
Loading weapon features.
Loading gun features.
```

7. Write a C++ program to add two numbers using templates.

```
#include<iostream>
using namespace std;
template<class T> T add (T &a,T &b)
{
 T result=a+b;
 return result;
}
int main ()
{
 int i=2;
 int j=3;
 float m=2.3;
 float n=1.2;
 cout<<" Enter two integer numbers:";
 cin>>i>>j;
 cout<<" Enter two real numbers";
 cin>>m>>n;
 cout<<"Addition of integer numbers:"<<add(i,j);
 cout<<'\n';
 cout<<"Addition of real numbers:"<<add(m,n);
 return 0;
}
```

Output:

```
Enter two integer numbers :2 3
 Enter two real numbers2.2 2.2
Addition of integer numbers :5
Addition of real numbers:4.4
```

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

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

class Student {
   string name;
   int age;

public:
   void setDetails(string n, int a) {
      name = n;
      age = a;
   }

   void display ( )
    {
      cout << "Name: " << name << ", Age: " << age << endl;
   }
     };

int main (  )
 {
   const int size = 3;

   // Array of pointers to Student objects
   Student* students[size];

   // Create Student objects dynamically and store their addresses
   for (int i = 0; i < size; ++i) {
      students[i] = new Student (); // allocate memory
      string name;
      int age;
      cout << "Enter name for student " << i + 1 << ": ";
      cin >> name;
      cout << "Enter age for student " << i + 1 << ": ";
      cin >> age;
      students[i]->setDetails (name, age);
   }

   cout << "\nStudent Details:\n";
   for (int i = 0; i < size; ++i) {
      students[i]->display ();  // Call function using pointer
   }

   // Free dynamically allocated memory
   for (int i = 0; i < size; ++i) {
```

```
    delete students[i];
  }

  return 0;
   }
```

Output:

```
Enter name for student 1: Prakash
Enter age for student 1: 14
Enter name for student 2: Sneha
Enter age for student 2: 26
Enter name for student 3: Sham
Enter age for student 3: 16

Student Details:
Name: Prakash, Age: 14
Name: Sneha, Age: 26
    Name: Sham, Age: 16
```

9. Write a C++ to demonstrate file reading and writing operations on files.

File Handling in C++

In C++, there are 3 file handling methods such as ifstream, ofstream and fstream. They are designed to manage the disk files. These are defined in fstream. That's why you have to include fstream whenever you are working with files in C++.

Files are mainly handled by three classes in C++: -

- **ofstream:-** Used to create files and write data into the files.
- **ifstream:**– Used to read information from the file.
- **fstream**:- Used to write data into file, read data from the file and also to create files.

You can perform the below operations through File Handling: -

- **open (): -** Create a file
- **read ():** – Reading file.
- **write ():** – Writing new data.
- **close ():** – Closing a file.

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

int main () {
  // Create a text file
  ofstream MyWriteFile("filename.txt");

  // Write to the file
  MyWriteFile << " KARNATAKA";

  // Close the file
  MyWriteFile.close();

  // Create a text string, which is used to output the text file
  string myText;

  // Read from the text file
  ifstream MyReadFile("filename.txt");

  // Use a while loop together with the getline() function to read the file line by line
  while (getline (MyReadFile, myText)) {
    // Output the text from the file
    cout << myText;
  }
  // Close the file
  MyReadFile.close();
}
```

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

In C++ Programming, the console IO operations are performed using the header file iostream.h. This header file provides two objects cin and cout to perform input and output operations respectively.
There are mainly two types of consol IO operations.

- **1. Unformatted consol IO -**
- **2. Formatted consol IO**

Unformatted console IO operations
We use the following built-in functions to perform operations of type unformatted consol IO operations.

⇢ get () and put ()

The **get( )** is a method of **cin** object used to input a single character from the standard input device (keyboard). Its main property is that it allows wide spaces and newline character. The get () function does not have any return value (void get ( )).

The **put()** is a method of **cout** object and it is used to print the specified character on standard output device (monitor).

```
#include <iostream>
using namespace std;
int main ()
{
   char ch;
   cout<<"Press any key: ";
   ch = cin.get();
   cout << "You have pressed one key: ";
   cout.put(ch);

   return 0;
}
```

Output:

Press any key:

You have pressed one key:

⇢ getline (char *buffer,int size) and write(char * buffer, int n)

The **getline (char *buffer,int size)** is a method of **cin** object and it is used to input a string with multiple spaces.

The **write (char * buffer, int n)** is a method of **cout** object and it is used to read n character from buffer variable.

```
#include <iostream>

using namespace std;

int main()

{     char ch[20];

    cout<<"What is your favourite website: ";

    cin.getline(ch, 20);

    cout <<endl<< "visit: www.";

    cout.write(ch, 17);

    cout<<".com"<<endl;

    return 0;

}
```

Output:

what is your favourite website: gat.com

visit: www.gat.com

VIVA VOCE QUESTIONS

Q1: What are literals?

A1: Literals are data items that never change their value during a program run.

Q2: What are five fundamental data types in C++?

A2 : char, int, float, double and void

Q3: What is reference?

A3: Reference is an alternative name for a variable.

Q4: What is a variable? A4: A variable is a named storage location. It has two associated values : rvalue that gives its contents and lvalue that gives its address.

Q5: What is class?

A5: A class is an expanded concept of a data structure: instead of holding only data, it can hold both data and functions.

Q6: What are the ways to define a class?

A6: While defining a class , its member functios can be either defined within the class definition or outside the class definition.

Q7: Are the inline functions called?

A7: Inline functions are not called , their code replaces their function –calls in the program.

Q8: What is friend function?

A8: A fiend function is a function which is not a member of a class but which is given special permission to access private and protected members of the class.

Q9: What is constructor?

A9 :A constructor is a member function with the same nme as that of its class and it is used to initialize the objects of that class type with a legel initial value.

Q10: If a class has no constructor, what does a compiler do?

A10 : If a class has no constructor defined , the compiler automatically generates one.

Q11: What is default constructor?

A11: Constructor with no arguments or all the arguments has default values.

Q12: What is inheritance? A1: Inheritance allows one class to reuse the state and behavior of another class. The derived class inherits the properties and method implementations of the base class and extends it by overriding methods and adding additional properties and methods.

Q13: What are the different forms of inheritance?

A13: Inheritance has many forms: single inheritance, multiple inheritance , hierarchical inheritance, multilevel inheritance, hybrid inheritance.

Q14: How can we inherit private member of a base class?

A14: To make a private member of a base class inheritable , declare it under protected section of base class.

Q15: What occurs in publicly derived class ?

A6: In a publicly derived class , the public and protected members remain public and protected.

Q15: What is Polymorphism?

A16: Polymorphism allows a client to treat different objects in the same way even if they were created from diffferent classes and exhibit different behaviors. You can use implementation inheritance to achieve polymorphism in languages such as C++ and Java. Base class object's pointer can invoke methods in derived class objects. You can also achieve polymorphism in C++ by function overloading and operator overloading.

Q17: What are the types of Polymorphism?

A17: There are two types of Polymorphism namely , compile time polymorphism and runtime polymorphism.

Q18: What is Operator overloading?

A18: When an operator is overloaded, it takes on an additional meaning relative to a certain class. But it can still retain all of its old meanings. Examples: 1) The operators >> and << may be used for I/O operations because in the header, they are overloaded. 2) In a stack class it is possible to overload the + operator so that it appends the contents of one stack to the contents of another. But the + operator still retains its original meaning relative to other types of data.
Q19: What does this pointers points to?

A19: this pointer points to the object for which this function was called.

Q20: What is Overriding?

A20: To override a method, a subclass of the class that originally declared the method must declare a method with the same name, return type (or a subclass of that return type), and same parameter list. The definition of the method overriding is: · Must have same method name. · Must have same data type. · Must have same argument list. Overriding a method means that replacing a method functionality in child class. To imply overriding functionality we need parent and child classes. In the child class you define the same method signature as one defined in the parent class.

Q21: What is a file?

A22: A file is a bunch of bytes stored on some storage device.

Q23: What does the fstream class do?

A23: The fstream class ties a file to the input stream for input; ofstream class ties a file to output stream for output; and fstream class ties a file to the stream for both input and output

Q24: How a file is closed?

A24: A file is closed i.e. its connection with the stream is terminated using function close ().

Q25: What does the fstream class do?

A5: The fstream class ties a file to the input stream for input; ofstream class ties a file to output stream for output; and fstream class ties a file to the stream for both input and output