C++ PROGRAMS:

1. Write a C++ Program to find area of triangle
2. Write a C++ Program to perform arithmetic operations using switch case
3. Write a C++ Program to check whether a number is even or odd
4. In C++, write a class called Employee () with properties like name, ID and salary and by Invoking them with multiple objects and implement the classes using default, parameterized and Copy constructor and destructor.

```
#include<iostream>
using namespace std;
class Employee{
  public:
  int id,salary;
  string empName;
  Employee(){
    cout<<"Enter the Id : ";
    cin>>id;
    cout<<"Enter the Salary : ";
    cin>>salary;
    cout<<"Enter the Name : ";
    cin>>empName;
  }
  Employee(Employee &a){
    id = a.id;
    salary = a.salary;
    empName = a.empName;
    cout<<"Id : "<<id<<endl;
    cout<<"Salary : "<<salary<<endl;
    cout<<"Name : "<<empName<<endl;
  }
  ~Employee(){
    cout<<"Object Terminated!!!";
  }
};
int main(){
  Employee a;
  Employee b = a;
  return 0;
}
```

5. Write the program using the constructor in CPP.

```

#include <iostream>
using namespace std;
class Employee {
		public:
		Employee() {
				cout<<"By Default Constructor Invoked"<<endl;
		}
};
```

```
int main() {
        Employee e1;
        Employee e2;
        return 0;
}
```

6. In C++, write a class called Sum() with variables (of any of these data types int, float and double) and invoke them with an object and implement the classes using Function Overloading [By changing number of arguments, by changing the Data types of arguments).

```
#include<iostream>
using namespace std;
class Sum{
  public:
  Sum(int a,int b){
    cout<<a+b<<endl;
  }
  Sum(float a,float b){
    cout<<a+b<<endl;
  }
  Sum(double a,double b,double c){
    cout<<a+b+c<<endl;
  }
};
int main(){
  Sum a(1,4);
  Sum b(2.43,5.67);
  Sum c(19.53652657,34.7328766,27.21321434);
  return 0;
}
```

7. Write a program using different type of constructor calls in CPP.

8. In C++ write a program to demonstrate the Macros by taking two variables and a macro MIN that takes these variables as argument and after that find their minimum and print the minimum using macro MIN ()

```
#include<iostream>
#define MIN(x,y) ((x<y)?x:y)
using namespace std;
int main() {
        cout<<MIN(4,3);
        return 0;
}
```

9. In C++, write two classes named Account (). Programmer () in C++ with user defined methods like salary [in Account ()]), bonus [in Programmer ()] and invoke them with an object and implement the classes using single-level inheritance.

   Note: Programmer) inherits Account ()]

10. In C++, write a program to implement Dynamic/Run time polymorphism using inheritance with a class named Base () using virtual destructor

    Note: [Derived() inherits Base())

```
#include<iostream>
using namespace std;
class Base {
        public:
                Base() {
                        cout << "\nConstructor Base class";
                }
                virtual ~Base() {
                        cout << "\nDestructor Base class";
                }
};
class Derived: public Base {
  public:
  Derived() {
        cout << "\nConstructor Derived class" ;
        }
        ~Derived() {
        cout << "\nDestructor Derived class";
        }
};
int main() {
        Base *p = new Derived;
        delete p;
}
```

11. In C++ Write three classes named Add1(), Add2(). Add3() and implement the concept of multiple inheritance by using user defined methods like get_a() in Add1(), get_b() in Add2() and display the result in method named display() in Add3().

```

#include<iostream>
using namespace std;
class Add1 {
        public:
        int a = 54;
        int get_a(){
                return a;
        }
};
class Add2 {
        public:
        int b = 44;
        int get_b(){
                return b;
        }
};
class Add3 : public Add1, Add2 {
        public:
        void display() {
                cout<<get_a()<<endl;
                cout<<get_b();
        }
};
int main() {
        Add3 obj;
        obj.display();
        return 0;
}
```

12. Write a CPP program using Template functions & classes in C++.

```
#include <iostream>
using namespace std;
template <class T> class Number {
  private:
  T num;
        T char;
  public:
  Number(T n) : num(n) {}
  T getNum() {
    return num;
  }
};
int main() {
   Number<int> numberInt(7);
   Number<double> numberDouble(7.7);
   cout << "int Number = " << numberInt.getNum() << endl;
   cout << "double Number = " << numberDouble.getNum() << endl;
   return 0;
}
```

13. In C++ write a program to demonstrate the usage of namespace by Creating two namespaces called First() and Second() with sayHello() function defined in both the namespaces and call sayHello() function that belongs to the namespace First and print the result

```
#include<iostream>
using namespace std;
namespace First {
        void sayHello() {
                cout << "Hello Earth-1..!\n";
        }
}
namespace Second {
        void sayHello() {
                cout << "Hello Earth-2..!\n";
        }
}
int main() {
        First::sayHello();
        Second::sayHello();
        return 0;
}
```

14. Write a CPP Program to show abstract class, pure abstract classes.

```
#include<iostream>
using namespace std;
class shape {
        public:
        virtual void draw() = 0;
};
class Rec : public shape {
        void draw(){
                cout<<"drawing Rectangle\n";
        }
};
class Circle : public shape {
        void draw() {
                cout<<"drawing Circle\n";
        }
};
int main() {
        Rec r;
        Circle c;
        shape *p1 = &r,*p2 = &c;
        p1->draw();
        p2->draw();
        return 0;
}
```

15. In C++, write a program to create an abstract class named Shape that contains two sub classes namely Rectangle() and Circle() such that each one of the classes extends the class Shape(). Each one of the classes contains only the method draw() that prints the name of the given class and invoke them by creating the objects.

```
#include<iostream>
using namespace std;
class shape {
        public:
        virtual void draw() = 0;
};
class Rec : public shape {
        void draw(){
                cout<<"drawing Rectangle\n";
        }
};
class Circle : public shape {
        void draw() {
                cout<<"drawing Circle\n";
        }
};
int main() {
        Rec r;
        Circle c;
        shape *p1 = &r,*p2 = &c;
        p1->draw();
        p2->draw();
        return 0;
}
```