1. WAP that asks for two integer numbers and performs sum operation in a function and returns its sum. [function]

```
#include<iostream>
using namespace std;
int Add(int num1,int num2)
{
   int sum=num1+num2;
   return sum;
}
int main()
{
   int num1,num2,sum;
   cout<<"Enter first number : ";
   cin>>num1;
   cout<<"\nEnter second number : ";
   cin>>num2;
   cout<<"\n"<<num1<<" + "<<num2<<" = "<<Add(num1,num2);
}
```

2. WAP using a function that calculates the factorial of a number. [recursion]

```
#include<iostream>
using namespace std;
int factorial(int num)
{
   if(num!=1)
   {
      return num*(num-1);
   }
   return 1;
}
int main()
{
   int num;
   cout<<"\nEnter the number : ";
   cin>>num;
   cout<<"\nThe factorial of "<<num<<" is "<<factorial(num);
}
```

3. WAP to ask for user name and store it as char datatype and display it. [char]

```
#include<iostream>
using namespace std;
class User{
   char name[20];
public :
   void getUser(){
   cout<<"\nEnter the User-Name : ";
   cin.get(name,20);
   }
   void displayUser(){
      cout<<"\nUser-Name = "<<name;
   }
};
int main()
{
   User user1;
   user1.getUser();
   user1.displayUser();
}
```

4. WAP having a value in a variable, store its address in a pointer variable and display address of both variables and value of variable using pointer. [pointer]

```
#include<iostream>
using namespace std;
int main()
{    int* ptr,var;
    cout<<"Enter a number : ";
    cin>>var;
    ptr=&var;
    cout<<"\nAddress of the variable = "<<ptr;
    cout<<"\nValue of the variable = "<<*ptr;
}
```

5. WAP to showcase the use of ‘this’ pointer. [this pointer]

Make a class with

*** private variables for a number and character. [values can be static]**

*** public functions to set the values and print the values.**

Create an object; using this object, access the function to store a number and character in private variables. Do note the argument

and parameter name must be same to use this pointer. Finally, print the variables.

```
#include<iostream>
using namespace std;
class Data{
   int num;
   char a;
public :
   void setdata(int num,char a)
   {       this->num=num;
           this->a=a;     }
   void displaydata()
   {   cout<<"\nNumber = "<<num;
       cout<<"\nCharacter = "<<a;   }
};
int main()  {
   Data obj1;
   obj1.setdata(55,'c');
   obj1.displaydata();
}
```

6. Inheritance example.

```
#include<iostream>
using namespace std;
class ParentClass{
   public :
   int var1 = 100;
};
class ChildClass : public ParentClass{
   public :
   int var2 = 500;
};
int main()
{
   ChildClass obj;
   cout<<obj.var2<<endl;
   cout<<obj.var1;
}
```

7. Polymorphism example.

```
#include<iostream>
using namespace std;
class Sum{
public:
    int add(int num1,int num2)
    {
        return num1 + num2;
    }
    int add(int num1,int num2,int num3)
    {
        return num1+num2+num3;
    }
};

int main()
{
    Sum obj;
    cout<<obj.add(10,20,30)<<endl;
    cout<<obj.add(11,22);
    return 0;  }
```

8. WAP using inheritance to demonstrate/print cat meows and has 4 legs in addition to printing it eats and sleeps which can be inherited from parent class.

```
#include<iostream>
using namespace std;
class Animal
{
public :
   void eat()
   {       cout<<"\nIt can Eat.";    }
   void sleep()
   {       cout<<"\nIt can Sleep.";    }
};
class Cat:public Animal
{    public :
   void meows()
   {       cout<<"\nIt meows.";    }
   void legs()
   {       cout<<"\nIt has 4 legs.";    }
};
```

```
int main()
{
    Cat cat1;
    cat1.meows();
    cat1.legs();
    cat1.eat();
    cat1.sleep();
}
```

9. WAP using polymorphism to demonstrate/print the sound of cat and dog having function name MakeSound(). Do note to use the parent class and child class. Create a base class named as Animal that has a function called MakeSound(). Derived classes of Animals can be Horses, Rabbits, Dogs, Cats, Frogs. They also have their own use of MakeSound function (the Horses neigh, dogs bark and the cat meows, etc.)

```
#include<iostream>
using namespace std;
class Animal{
public :
   void Makesound()
   {       cout<<"\nThe Cat Meows.";
          cout<<"\nThe Dog Barks.";      }
};
class Horses : public Animal{
   public :
   void Makesound()
   {
      cout<<"\nThe Horses Neigh.";
   }
} ;
```

```
class Rabbit : public Animal{
    public :
    void Makesound()
    {       cout<<"\nThe Rabbit squeaks.";     }
};
class Frogs : public Animal{
    public :
    void Makesound()
    {       cout<<"\nThe Frog croaks.";     }
};
int main() {
    Animal animal1;
    Horses horse1;
    Rabbit rabbit1;
    Frogs frog1;

    animal1.Makesound();
    horse1.Makesound();
    rabbit1.Makesound();
    frog1.Makesound();
}
```