Task 1:

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

class Car{
    private:
        string engine;
        int currentSpeed;
        string fuel;
    public:
        Car(string eng,int speed,string fuel):fuel(fuel),engine(eng),currentSpeed(speed){
        }
        void startEngine(){
            engine="Start";
        }
        void stopEngine(){
            engine="Stop";
        }
        void setSpeed(int speed){
            currentSpeed=speed;
        }
        int getSpeed(){
            return currentSpeed;
        }
        void display(){
            cout<<"Engine: "<<engine<<" Current Speed: "<<currentSpeed<<" Fuel level :"<<fuel<<endl;
        }
};
int main(){
    Car c1("start",30,"below average");
    c1.display();
    c1.setSpeed(120);
    c1.startEngine();
    c1.display();
}
```

```
Engine: start Current Speed: 30 Fuel level :below average
Engine: Start Current Speed: 120 Fuel level :below average

--------------------------------
Process exited after 0.2804 seconds with return value 0
Press any key to continue . . .
```

Task # 02:

Encapsulation:

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

class bankAccount{
    private:
        double balance;
    public:
        void setBalance(double stat){
            balance=stat;
        }
        double getBalance(){
            return balance;
        }
        void deposit(double amount){
            balance+=amount;
        }
        void withdraw(double with){
            balance-=with;
        }
};

int main(){
    bankAccount b1;
    b1.setBalance(321.12);
    cout<<"Balance: "<<b1.getBalance()<<endl<<endl;
    b1.deposit(231.1);
    b1.withdraw(111);
    cout<<"Balance: "<<b1.getBalance()<<endl<<endl;
}
```

```
Balance: 321.12

Balance: 441.22


--------------------------------
Process exited after 0.2482 seconds with return value 0
Press any key to continue . . .
```

Abstraction:

```
#ifndef BANKACCOUNT
#define BANKACCOUNT
class BankAccount {
  private:
    double balance;
  public:
    BankAccount(double initial_balance);
    void deposit (double amount);
    withdraw(double amount);
    double getBalance() const;
};
#endif
```

```
#include "BankAccount.h"
#include <iostream>

using namespace std;

BankAccount::BankAccount(double initial_balance) : balance(initial_balance) {}

void BankAccount::deposit(double amount) {
    if (amount > 0) {
        balance += amount;
}
}

bool BankAccount::withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        return true;
    } else {
        cout << "Insufficient funds" << endl;
        return false;
    }
}

double BankAccount::getBalance() {
    return balance;
}
```

Task # 03:

Calculator.h

```
#ifndef CALCULATOR
#define CALCULATOR

class Calculator{
    private:
        int a;
        int b;
    public:
        Calculator(int a,int b);
        void add();
        void subtract();
        void multiply();
        void divide();
};
#endif
```

Calculator.cpp

```
#ifndef CALCULATOR
#define CALCULATOR

class Calculator{
    private:
        int a;
        int b;
    public:
        Calculator(int a,int b);
        void add();
        void subtract();
        void multiply();
        void divide();
};
#endif
```

```
#include <iostream>
#include "Calculator.h"
using namespace std;
int main(){
    Calculator c1(10,5);
    c1.add();
    c1.subtract();
    c1.divide();
    c1.multiply();
}
```

Task # 04:

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

class Animal{
    public:
        virtual void makesound()=0;
};

class Dog:public Animal{
    public:
        void makesound(){
            cout<<"Dog Barks"<<endl;
        }
};

class Cat:public Animal{
    public:
        void makesound(){
            cout<<"Meow meow"<<endl;
        }
};

class Cow:public Animal{
    public:
        void makesound(){
            cout<<"mooo mooo"<<endl;
        }
};

int main(){
    Animal *animal[3];

    Cat c1;
    Dog d1;
    Cow cow;

    animal[0]=&c1;
    animal[1]=&d1;
    animal[2]=&cow;

    for(int i=0;i<3;i++){
        animal[i]->makesound();
```

```
Meow meow
Dog Barks
mooo mooo


--------------------------------
Process exited after 0.2693 seconds with return value 0
Press any key to continue . . .
```

Task # 05:

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

class Library {
private:
    string title;
    string author;
    bool availability;

public:
    int flag;

    void setTitle() {
        if (flag == 1) {
            cout << "Enter title: ";
            cin.ignore();
            getline(cin, title);
        } else {
            title = "";
        }
    }

    void setAuthor() {
        if (flag == 1) {
            cout << "Enter author: ";
            getline(cin, author);
        } else {
            author = "";
        }
    }

    void setAvail() {
        if (flag == 1) {
            int status;
            cout << "If available press '1' or if not then press '0': ";
            cin >> status;
            availability = (status == 1);
```

```
    }

    string getTitle() {
        return title;
    }

    bool getAvailable() {
        return availability;
    }

    virtual void add() = 0;
    virtual void remove() = 0;
    virtual void borrow() = 0;
};

class Librarian : public Library {
public:
    void add() {
        flag = 1;
        setTitle();
        setAuthor();
        setAvail();
        cout << "Book added successfully." << endl;
    }

    void remove() {
        flag = 0;
        setTitle();
        setAuthor();
        setAvail();
        cout << "Book removed." << endl;
    }

    void borrow() {}
};

class User : public Library {
public:
    void add() {}
```

```
    void remove() {}

    void borrow() {
        string choice;
        cout << "Enter book title: ";
        cin.ignore();
        getline(cin, choice);

        if (getTitle() == choice) {
            if (getAvailable()) {
                cout << "Book is issued." << endl;
            } else {
                cout << "Book is not available." << endl;
            }
        } else {
            cout << "Book not found." << endl;
        }
    }
};

int main() {
    Library* lib1;
    Library* lib2;
    Librarian l1;
    User u1;

    lib1 = &l1;
    lib2 = &u1;

    lib1->add();
    lib2->borrow();
    lib1->remove();
    lib2->borrow();

    return 0;
}
```

```
Enter title: rayyan
Enter author: rayyan
If available press '1' or if not then press '0': 1
Book added successfully.
Enter book title: Rrrra
Book not found.
Book removed.
Enter book title: aaaa
Book not found.

--------------------------------
Process exited after 14.61 seconds with return value 0
Press any key to continue . . .
```

Task # 06:

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

class Shape{
    public:
        virtual void area()=0;
};

class Circle:public Shape{
    private:
        int radius;
    public:
        Circle(int rad):radius(rad){}
        void area(){
            cout<<"Area of circle with radius "<<radius<<" is "<<3.14*radius*radius<<endl;
        }
};

class Rectangle:public Shape{
    private:
        int lenght;
        int breadth;
    public:
        Rectangle(int len,int bread):lenght(len),breadth(bread){
        }
        void area(){
            cout<<"Area of Rectangle with lenght and width "<<lenght<<"x"<<breadth<<" is "<<breadth*lenght<<endl;
        }
};

class Triangle:public Shape{
    private:
        int base;
        int height;
    public:
        Triangle(int b,int h):height(h),base(b){}
        void area(){
            cout<<"Area of Triangle with base and height "<<base<<"x"<<height<<" is "<<0.5*base*height<<endl;
        }
};


int main(){
    Shape *s1[3];
    Circle c1(3);
    Rectangle r1(2,4);
    Triangle t1(3,6);
    s1[0]=&c1;
    s1[1]=&r1;
    s1[2]=&t1;
    for(int i=0;i<3;i++){
        s1[i]->area();
    }
}
```

```
Area of circle with radius 3 is 28.26
Area of Rectangle with lenght and width 2x4 is 8
Area of Triangle with base and height 3x6 is 9

--------------------------------
Process exited after 0.2554 seconds with return value 0
Press any key to continue . . .
```