Assignment 4

Q1 :

```
#include<iostream>
#include<string.h>
using namespace std;
// *** Employe  ( Base class )***
class Employee{
    public :
// Attributes of Employee :
    int eId;
    char eName[30];
    double eSalary;
        Employee(){
        this->eId=0;
        strcpy(this->eName,"Null");
        this->eSalary=0;
    }
// Methods of Employee :
    //1. Parameterized Constructor
    Employee(int eId,const char* eName,double eSalary){
        this->eId=eId;
        strcpy(this->eName,eName);
        this->eSalary=eSalary;
    }
    // 2.Setter
    void setId(int eId){
        this->eId=eId;
    }
    void setName(const char* eName){
        strcpy(this->eName,eName);
    }
    void setSalary(double eSalary){
        this->eSalary=eSalary;
    }
    // 3. Getter
    int getId(){
        return this->eId;
    }
    char* getName(){
        return this->eName;
    }
    double getSalary(){
        return this->eSalary;
    }
    // 4. Display Employee
    virtual void display(){
```

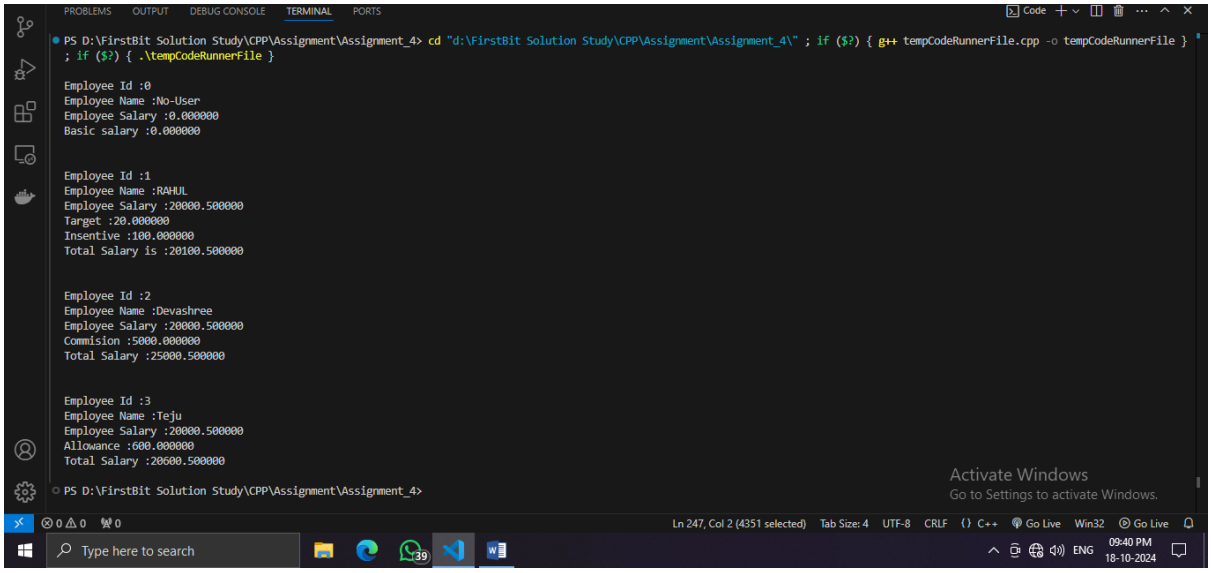
```
        printf("\nEmployee Id :%d",this->eId);
        printf("\nEmployee Name :%s",this->eName);
        printf("\nEmployee Salary :%lf",this->eSalary);
    }
    // 5. Calculate salary
    virtual double calSalary(){
        printf("\nBasic salary :%lf\n\n",this->eSalary);
    }
};
// *** HR  ***
class HR:public Employee{
    public :
    // Additional Attribute
    double commision;
    // Additional Method
    // 1.Default Constructor
    HR(){
        this->commision=0;
    }
    // 2. Parameterized Constructor
    HR(int eId,const char* eName,double eSalary,double
commision):Employee(eId,eName,eSalary){
        this->commision=commision;
    }
    // 3. Setters
    void setCommision(double commision){
        this->commision=commision;
    }
    // 4. Getters
    double getCommision(){
        return this->commision;
    }
    // 5. Display Information
    virtual void display(){
        Employee::display();
        printf("\nCommision :%lf",this->commision);
    }
    virtual double calSalary(){
        printf("\nTotal Salary :%lf\n\n",this->eSalary+this->commision);
        return this->eSalary+this->commision;
    }
};

// *** Sales Manager ***
class SalesManager:public Employee{
    public :
    // Additional Attribute :
    double target;
```

```
    double insentive;
    // Additional Method :
    //1. Default Constructor
    SalesManager(){
        this->target=0;
        this->insentive=0;
    }
    // 2. Parametrized constructor
    SalesManager(int eId,const char* eName,double eSalary,double target,double
insentive):Employee(eId,eName,eSalary){
        this->target=target;
        this->insentive=insentive;
    }
    // 3.Setters
    void setTarget(double target){
        this->target=target;
    }
    void setInsentive(double insentive){
        this->insentive=insentive;
    }
    //4.Getters
    double getTarget(){
        return this->target;
    }
    double getInsentive(){
        return this->insentive;
    }
    // 5.Display Information
    virtual void display(){
        Employee::display();
        printf("\nTarget :%lf",this->target);
        printf("\nInsentive :%lf",this->insentive);
    }
    // 6. Calculate Salary
    virtual double calSalary(){
        printf("\nTotal Salary is :%lf\n\n",this->eSalary+this->insentive);
        return this->eSalary+this->insentive;
    }
};
// *** Admin ***
class Admin:public Employee{
    public:
    // Additional Attribute
    double allowance;
    // Additional Method
    // 1. Default Constructor
    Admin(){
        this->allowance=0;
```

```
    }
    // 2.Parameterized Constructor
    Admin(int eId,const char* eName,double eSalary,double
allowance):Employee(eId,eName,eSalary){
        this->allowance=allowance;
    }
    // 3.Setters
    void setAllowance(double allowance){
        this->allowance=allowance;
    }
    // 4.Getters
    double getAllowance(){
        return this->allowance;
    }
    // 5.Display Info
    virtual void display(){
        Employee::display();
        printf("\nAllowance :%lf",this->allowance);
    }
    virtual double calSalary(){
        printf("\nTotal Salary :%lf\n\n",this->eSalary+this->allowance);
        return this->eSalary+this->allowance;
    }

};

int main(){
    // Creating Array of Employee to store all
    Employee e1(000,"No-User",00.000);
    e1.display();
    e1.calSalary();

    // Sales Manager
    SalesManager s1(1,"RAHUL",20000.500,20.0,100.0);
    s1.display();
    s1.calSalary();

    //HR
    HR h1(2,"Devashree",20000.500,5000.00);
    h1.display();
    h1.calSalary();

    //Admin
    Admin a1(3,"Teju",20000.500,600.00);
    a1.display();
    a1.calSalary();

// p[0]=new SalesManager(101,"xyz",10000.0,153,1000);
```

```
// p[1]=new HR(102,"abc",10000.0,2000);
// p[2]=new Admin(103,"pqr",10000,1200);


return 0;
}
```

Output :-



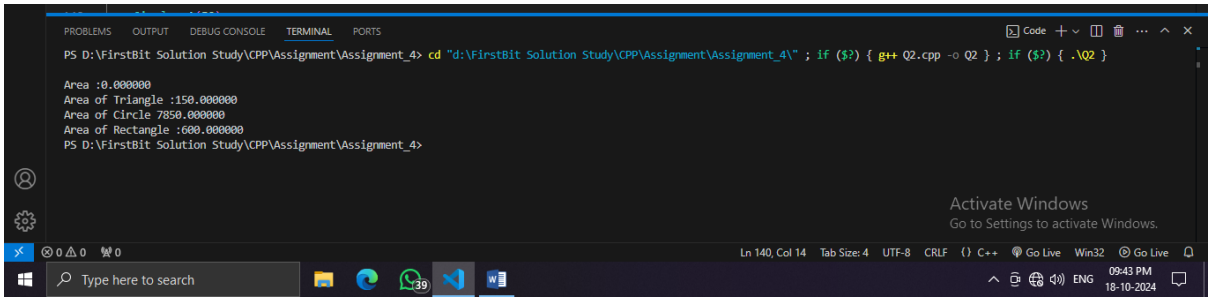
Q2 :-

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

class Shape{
    public :
    // Attribute :-->
    double area;
    //Method :-->
    // 1.Default Constructor
    Shape(){
        this->area=0;
    }
    // 2.Parameter constructor
    Shape(double area){
        this->area=area;
    }
    // 3.calculate area
    virtual double calArea(){
        return this->area;
    }
    // 4.Display Area
```

```
    virtual void display(){
        printf("\nArea :%lf",this->area);
    }
};
class Triangle:public Shape{
    public:
    int height;
    int base;
    //Method :
    //1. Default Constructor
    Triangle(){
        this->height=0;
        this->base=0;
    }
    // 2. Parameter Constructor
    Triangle(int height ,int base){
        this->height=height;
        this->base=base;
    }
    virtual double calArea(){
         this->area=(0.5*this->base*this->height);
        return area;
    }
    virtual void display(){
        printf("\nArea of Triangle :%lf",this->area);
    }
};
class Circle:public Shape{
    public:
    //Additional Attributes
    int radius;
    // Methods
    //1.Default Constructor
    Circle(){
        this->radius=0;
    }
    //2. Parameter Constructor
    Circle(int radius){
        this->radius=radius;
    }
    //3.Calculate area
    virtual double calArea(){
        this->area=(3.14*this->radius*this->radius);
        return area;
    }
    virtual void display(){
        printf("\nArea of Circle %lf",this->area);
    }
```

```
};

class Rectangle:public Shape{
    public:
    //Additional Attributes
    int len;
    int bre;
    //Method
    // 1. default Constructor
    Rectangle(){
        this->len=0;
        this->bre=0;
    }
    Rectangle(int len,int bre){
        this->len=len;
        this->bre=bre;
    }
    //3.Calculate area
    virtual double calArea(){
        this->area=this->len*this->bre;
        return area;
    }
    //4.Display
    virtual void display(){
        printf("\nArea of Rectangle :%lf",this->area);
    }
};

int main(){
    Shape s1;
    s1.calArea();
    s1.display();

    Triangle t1(15,20);
    t1.calArea();
    t1.display();

    Circle c1(50);
    c1.calArea();
    c1.display();

    Rectangle r1(20,30);
    r1.calArea();
    r1.display();
return 0;
}
```

Output :

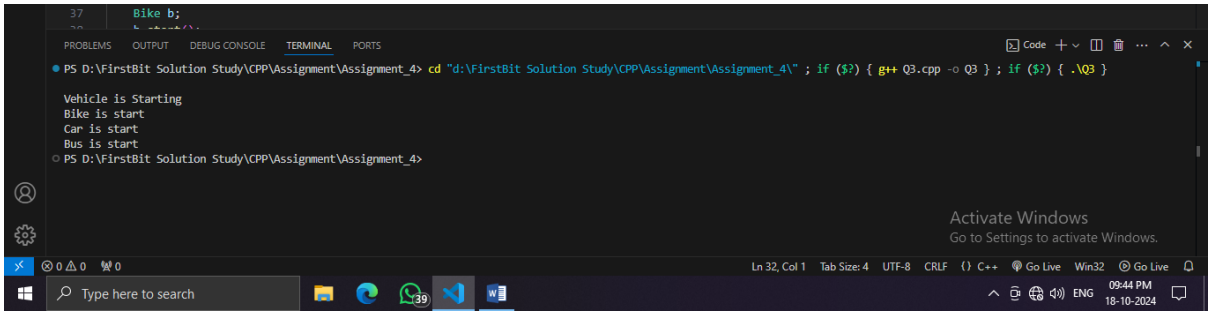


Q 3 :

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

class Vehicle{
    public:
    virtual void start(){
        printf("\nVehicle is Starting");
    }
};
class Bike:public Vehicle{
    public:
    void start(){
        printf("\nBike is start");
    }
};
class Car:public Vehicle{
    public:
    void start(){
        printf("\nCar is start");
    }
};
class Bus:public Vehicle{
    public:
    void start(){
        printf("\nBus is start");
    }
};

int main(){
    Vehicle v;
    v.start();

    Bike b;
    b.start();

    Car c;
    c.start();
```

```
    Bus bus;
    bus.start();
    return 0;
}
```

Output :-



Q4 :

```
/*
4. Write  more code to show inheritance on your own
*/
#include<iostream>
#include<string.h>
using namespace std;

class Person{
    public:
    // Attribute
    char Name[10];
    int age;
    // Method
    //1. Default Constructor
    Person(){
        strcpy(this->Name,"Not Null");
        this->age=0;
    }
    //2.Parameter Constructor
    Person(const char* Name,int age){
        strcpy(this->Name,Name);
        this->age=age;
    }
    //3.Display Information
    virtual void display(){
        printf("\nName :%s",this->Name);
        printf("\nAge :%d\n",this->age);
    }
```

```cpp
};

class Student :public Person{
    public:
    //Attribute
    int roll;
    double marks;
    //Method
    //1.Default
    Student(){
        this->roll=0;
        this->marks=0;
    }
    //2.Parameter
    Student(const char* Name,int age,int roll,double marks):Person(Name,age){
        this->roll=roll;
        this->marks=marks;
    }
    //3.Display
    void display(){
        printf("\nRoll Id:%d",this->roll);
        printf("\nMarks :%lf",this->marks);
        Person::display();
    }
};

int main(){
    Person p1("Rahul",25);
    p1.display();

    Student s1("Rahul",25,1,89.44);
    s1.display();
}
```

Output :