```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <cstdio>
#include <iostream>
#include <iomanip>
using namespace std;

struct Employee
{
   int id;
   char name[20];
   double salary;

   // default
   Employee()
   {
      this->id = 0;
      strcpy(this->name, "NotGiven");
      this->salary = 0;
   }
   // parameterised Constructor
   Employee(int id, const char *name, double salary)
   {
      this->id = id;
      strcpy(this->name, name);
      this->salary = salary;
   }

   // setters
```

```
void setId(int id)
{
   this->id = id;
}
void setName(const char *name)
{
   strcpy(this->name, name);
}
void setSalary(double salary)
{
   this->salary = salary;
}

// gettters
int getId()
{
   return this->id;
}
char *getName()
{
   return this->name;
}
double getSalary()
{
   return this->salary;
}

double calculateSalary()
{
   return this->salary;
}
```

```
    void display()
    {
        cout << "\nId         = " << this->id;
        cout << "\nName       = " << this->name;
        cout << "\nSalary     = " << this->salary;
    }
};
struct SalesManager : public Employee
{
    double Incentive;
    int target;

    // setter
    void setIncentive(double incentive)
    {
        this->Incentive = incentive;
    }
    void setTarget(int target)
    {
        this->target = target;
    }

    // getters
    double getIncentive()
    {
        return this->Incentive;
    }
    int getTarget()
    {
        return this->target;
    }
```

```
   // default
   SalesManager() : Employee()
   {
      this->Incentive = 0;
      this->target = 0;
   }
   // parameterised
   SalesManager(int id,const char *name, double salary, double incentive, int target) : Employee(id,
name, salary)
   {
      this->Incentive = incentive;
      this->target = target;
   }
   double calculateSalary()
   {
      return this->Incentive + this->getSalary();
   }
   void display()
   {
      Employee::
         display();
      cout << "\nIncentive   = " << this->Incentive;
      cout << "\nTarget      = " << this->target;
   }
};
struct Admin : public Employee
{
   double allowance;

   void setAllowance(double allowance)
```

```
    {
        this->allowance = allowance;
    }

    double getAllowance()
    {
        return this->allowance;
    }

    Admin() : Employee()
    {
        this->allowance = 0;
    }
    Admin(int id,const char *name, double salary, double allowance) : Employee(id, name, salary)
    {
        this->allowance = allowance;
    }
    double calculateSalary()
    {
        return this->allowance + this->getSalary();
    }
    void display()
    {
        Employee::display();
        cout << "\nAllowance = " << this->allowance;
    }
};
struct HR : public Employee
{
    double commision;
```

```
  void setCommision(double commision)
  {
    this->commision = commision;
  }

  double getCommision()
  {
    return this->commision;
  }

  HR() : Employee()
  {
    this->commision = 0;
  }
  HR(int id,const char *name, double salary, double commision) : Employee(id, name, salary)
  {
    this->commision = commision;
  }
  double calculateSalary()
  {
    return this->commision + this->getSalary();
  }

  void display()
  {
    Employee::
      display();
    cout << "\nCommision   = " << this->commision;
  }
};
```

```
int main()
{
  SalesManager s1(101,"suraj",34000,4500,200);
  HR h1(102,"Nikhil",34000,4500);
  Admin a1(103,"vivek",45000,4500);

  s1.display();
  cout<<"\n";
  h1.display();
  cout<<"\n";
  a1.display();

  cout<<s1.calculateSalary();
  cout<<"\nTotal Salary = "<<h1.calculateSalary();

}
```

```
Name        = suraj
Salary      = 34000
Incentive   = 4500
Target      = 200

Id          = 102
Name        = Nikhil
Salary      = 34000
Commision   = 4500

Id          = 103
Name        = vivek
Salary      = 45000
Allowance = 450038500
Total Salary = 38500
```

```
#include <iostream>
using namespace std;
float PI = 3.147;
struct Shape
{
   float area;
};
struct circle : public Shape
{
   float r;

   void setRadius(float r)
   {
      this->r = r;
   }
   float getRadius()
   {
      return this->r;
   }
   circle()
   {
      this->r = 0;
   }
   circle(float r)
   {
      this->r = r;
   }
   void display()
   {
      cout << "radius  = " << this->r;
   }
```

```
    float calculateArea()
    {
        return this->area = PI * this->r * this->r;
    }
};
struct tringle : public Shape
{
    float b, h;

    void setBredth(float b)
    {
        this->b = b;
    }
    void setHeight(float h)
    {
        this->h = h;
    }
    float getBredth()
    {
        return this->b;
    }
    float getHeigth()
    {
        return this->h;
    }

    tringle()
    {
        this->b = 0;
        this->h = 0;
    }
```

```
    tringle(float b, float h)
    {
        this->b = b;
        this->h = h;
    }
    void display()
    {
        cout << "\nBredth  = " << this->b;
        cout << "\nHeigth  = " << this->h;
    }
    float calculateArea()
    {
        return this->b * this->h;
    }
};
struct rectangle : public Shape
{
    float l, w;
    rectangle()
    {
        this->l = 0;
        this->w = 0;
    }
    rectangle(float l, float w)
    {
        this->l = l;
        this->w = w;
    }
    void setLength(float l)
    {
        this->l = l;
```

```
    }
    void setWidth(float w)
    {
        this->w = w;
    }
    float getLength()
    {
        return this->l;
    }
    float getWidth()
    {
        return this->w;
    }

    void display()
    {
        cout << "\nLenth  = " << l;
        cout << "\nWidth  = " << w;
    }
    float calculateArea()
    {
        area = 2 * l * w;
        return area;
    }
};
int main()
{
    circle c1(56);
    rectangle r1(10, 20);
    tringle t1(30, 40);
```

```
    c1.display();
    cout << "\nArea of Circle = " << c1.calculateArea();
    cout << "\n";
    r1.display();
    cout << "\nArea of Rectangle = " << r1.calculateArea();
    cout << "\n";
    t1.display();
    cout << "\nArea of Tringle = " << t1.calculateArea();
}
```

```
radius  = 56
Area of Circle = 9868.99

Lenth  = 10
Width  = 20
Area of Rectangle = 400

Bredth  = 30
Heigth  = 40
Area of Tringle = 1200
```

```
// 3. Write a code to implement inheritance where vehicle is base class and derived
// classes like bike, car, bus etc.

#include <iostream>
using namespace std;

struct vehicle
{
   int noOfWheels;

   vehicle()
   {
      this->noOfWheels=0;
   }
   vehicle(int wheel)
   {
      this->noOfWheels=wheel;
   }
   void setWheels(int wheel)
   {
      this->noOfWheels = wheel;
   }
   int getWheels()
   {
      return this->noOfWheels;
   }
   void display()
   {
      cout<<"\n Vehicle Info";
      cout << "\nNo of wheels = " << this->noOfWheels;
   }
```

```
};
struct bus : public vehicle
{
   int noofWindow;
   bus():vehicle()
   {
      this->noofWindow=0;
   }
   bus(int wheel, int window):vehicle(wheel)
   {
      this->noofWindow=window;
   }
   void setWindows(int window)
   {
      this->noofWindow=window;
   }
   int getWindows()
   {
      return this->noofWindow;
   }
   void display()
   {
      vehicle::display();
      cout<<"\n Bus Info";
      cout<<"\nNo of Windows = "<<this->noofWindow;
   }
};
struct car : public vehicle
{
   int noofWindow;
   car():vehicle()
```

```
    {
        this->noofWindow=0;
    }
    car(int wheel, int window):vehicle(wheel)
    {
        this->noofWindow=window;
    }
    void setWindows(int window)
    {
        this->noofWindow=window;
    }
    int getWindows()
    {
        return this->noofWindow;
    }
    void display()
    {
        vehicle::display();
        cout<<"\n Car Info";
        cout<<"\nNo of Windows = "<<this->noofWindow;
    }
};
struct bike : public vehicle
{
    int noofShockups;
    bike():vehicle()
    {
        this->noofShockups=0;
    }
    bike(int wheel,int shockup):vehicle(wheel)
    {
```

```
        this->noofShockups=shockup;
    }
    void setShockup(int shockup)
    {
        this->noofShockups=shockup;
    }
    int getShockup()
    {
        return this->noofShockups;
    }
    void display()
    {
        vehicle::display();
        cout<<"\n Bike Info";
        cout<<"\nNo of Shockups = "<<this->noofShockups;
    }
};
int main()
{
    vehicle v1(10);
    v1.display();

    bus b1(10,20);
    b1.display();

    car c1(4,4);
    c1.display();

    bike bk1(2,6);
    bk1.display();
}
```

```
 Vehicle Info
No of wheels = 10
 Vehicle Info
No of wheels = 10
 Bus Info
No of Windows = 20
 Vehicle Info
No of wheels = 4
 Car Info
No of Windows = 4
 Vehicle Info
No of wheels = 2
 Bike Info
No of Shockups = 6
```

```
//mouse heirarchy

#include<iostream>
#include<string.h>
using namespace std;

class mouse
{
   protected:
   int productId;
   public:
   mouse()
   {
      this->productId=0;
   }
   mouse(int id)
   {
      this->productId=id;
   }
   void setProductId(int id)
   {
      this->productId=id;
   }
   int getProductId()
   {
      return this->productId;
   }
   void display()
   {
      cout<<"\nProductId = "<<this->productId;
   }
```

```
};
class opticalMouse:public mouse
{
   protected:
   const char *sensorType;
   public:
   opticalMouse():mouse()
   {
      // strcpy(this->sensorType,"NotGiven");
      this->sensorType="NotGiven";
   }
   opticalMouse(int id,const char* sensorType):mouse(id)
   {
      // strcpy(this->sensorType,sensorType);
      this->sensorType=sensorType;
   }
   void setSensorType(const char* sensorType)
   {
      // strcpy(this->sensorType,sensorType);
      this->sensorType=sensorType;
   }
   const char* getSensorType()
   {
      return sensorType;
   }
   void display()
   {
      mouse::display();
      cout<<"\nSensorType = "<<this->sensorType;
   }
```

```
};
class ballMouse:public mouse
{
   protected:
   const char *ballType;

   public:
   ballMouse():mouse()
   {
      // strcpy(this->ballType,"NotGiven");
      this->ballType="NotGiven";
   }
   ballMouse(int id,const char* ballType):mouse(id)
   {
      // strcpy(this->ballType,ballType);
      this->ballType=ballType;
   }
   void setBallType(const char* ballType)
   {
      // strcpy(this->ballType,ballType);
      this->ballType=ballType;
   }
   const char* getBallType()
   {
      return this->ballType;
   }
   void display()
   {
      mouse::display();
      cout<<"\nBallType = "<<this->ballType;
   }
```

```
};
int main()
{
    const char *sensor="Laser";
    const char* ballType="Rubber";
    opticalMouse op1;
    op1.setProductId(101);
    op1.setSensorType(sensor);
    op1.display();

    ballMouse b1(102,ballType);
    cout<<"\n";
    b1.display();

    return 0;
}
```

```
ProductId = 101
SensorType = Laser

ProductId = 102
BallType = Rubber
```

```
// artist painter musician

#include <iostream>
#include <string.h>
using namespace std;

class artist
{
protected:
   char name[20];
   int age;
   char gender[10];

public:
   // Default constructor suru
   artist()
   {
      strcpy(this->name, "NotGiven");
      this->age = 0;
      strcpy(this->gender, "NotDefine");
   }
   // Parameterised constructor suru
   artist( char *name, int age,  char *gender)
   {
      strcpy(this->name, name);
      this->age = age;
      strcpy(this->gender, gender);
      // this->gender=gender;
   }
   // setters
   void setName( char *name)
```

```
{
   strcpy(this->name, name);
   // this->name=name;
}
void setAge(int age)
{
   this->age = age;
}
void setGender( char *gender)
{
   strcpy(this->gender, gender);
   // this->gender=gender;
}
// getters
char *getName()
{
   return this->name;
}
int getAge()
{
   return this->age;
}
char *getGender()
{
   return this->gender;
}

// Display
void display()
{
   cout << "\nName   = " << this->name;
```

```
        cout << "\nAge    = " << this->age;
        cout << "\nGender = " << this->gender;
    }
};
class Painter : public artist
{
protected:
    int noOfBrush;
    char paintingType[20];

public:
    Painter() : artist()
    {
        this->noOfBrush = 0;
        strcpy(this->paintingType, "NotSpecified");
        // this->paintingType=
    }
    Painter( char *name, int age,  char *gender, int noOfBrush,  char *paintingType) : artist(name, age, gender)
    {
        this->noOfBrush = noOfBrush;
        strcpy(this->paintingType, paintingType);
        // this->paintingType=paintingType;
    }
    void setBrush(int noOfBrush)
    {
        this->noOfBrush = noOfBrush;
    }
    void setPaintingType(const char *paintingType)
    {
        strcpy(this->paintingType, paintingType);
```

```
        // this->paintingType=paintingType;
    }
    int getBrush()
    {
        return this->noOfBrush;
    }
    char *getPaintingType()
    {
        return this->paintingType;
    }

    void display()
    {
        artist::display();
        cout << "\nNoOfBrush    = " << this->noOfBrush;
        cout << "\nPaintingType = " << this->paintingType;
    }
};
int main()
{
    artist a1;
    char name[10] = "Vivek";
    char gender[10] = "Male";
    char name1[10] = "Shubham";
    char paintingType[20] = "OilPainting";
    Painter p1(name, 81, gender, 5, paintingType);
    a1.setAge(27);
    a1.setName(name1);
    a1.setGender(gender);
    a1.display();
    p1.display();
```

}

```
Name   = Shubham
Age    = 27
Gender = Male
Name   = Vivek
Age    = 81
Gender = Male
NoOfBrush    = 5
PaintingType = OilPainting
```