Name - Akriti Choudhary
Roll number - 2005776
Lab11
Subject - OOP lab
Class - B14
Branch - CSE
Date- 11/11/2021

Question 1) Create a class shape. Derive three classes from it; Circle, Square and Triangle. Find area of each shape and display it, using virtual function.

```
#include <iostream>
#include <cstring>
using namespace std;
class Shape
{
public:
   virtual void input()
   {
      cout << "Shape's input called";
   }
   virtual void area()
   {
      cout << "Shape's area";
   }
};
class Circle : public Shape
{
   int radius;

public:
   void input()
   {
      cout << "Enter radius of circle: ";
      cin >> radius;
   }
   void area()
   {
      cout << "\nArea of Circle is:" << 3.14 * radius * radius << endl;
   }
};
class Triangle : public Shape
{
   int b, h;

public:
   void input()
   {
      cout << "Enter base of triangle: ";
      cin >> b;
      cout << "Enter height of triangle: ";
      cin >> h;
   }
   void area()
   {
      cout << "Area of triangle is: " << 0.5 * h * b << endl;
   }
};
class square : public Shape
{
   int l;
```

```
public:
   void input()
   {
      cout << "Enter the side of square : ";
      cin >> l;
   }
   void area()
   {
      cout << "Area of square is: " << l * l << endl;
   }
};
int main()
{
   Shape *p1, *p2, *p3;
   Circle c;
   Triangle t;
   square r;
   p1 = &c;
   p2 = &t;
   p3 = &r;
   p1->input();
   p2->input();
   p3->input();
   p1->area();
   p2->area();
   p3->area();
}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\11_11_2021> g++ area.cpp -oarea
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\11_11_2021> ./area
Enter radius of circle: 2
Enter base of triangle: 4
Enter height of triangle: 6
Enter the side of square : 3

Area of Circle is:12.56
Area of triangle is: 12
Area of square is: 9
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\11_11_2021>
```

Question 2) Create a class which stores employee name,id and salary Derive two classes from 'Employee' class: 'Regular' and 'Part-Time'. The 'Regular' class stores DA, HRA and basic salary. The 'Part-Time' class stores the number of hours and pay per hour. Calculate the salary of a regular employee and a par-time employee, using virtual function.

```
#include <iostream>
using namespace std;
class employee
{
public:
   char name[25];
   int id, salary, DA, HRA, hr, pph;
   void info()
   {
      cout << "Enter name : ";
      cin >> name;
      cout << "Enter ID : ";
      cin >> id;
   }
   void regular()
   {
      cout << "Enter salary : ";
      cin >> salary;
      cout << "Enter DA : ";
      cin >> DA;
      cout << "Enter HRA : ";
      cin >> HRA;
   }
   void part()
   {
      cout << "Enter number of hours : ";
      cin >> hr;
      cout << "Enter pay per hour : ";
      cin >> pph;
   }
   virtual void sal() = 0;
};
class regular : public employee
{
public:
   void sal()
   {
      cout << "\nSalary of regular employee : " << salary + DA + HRA << endl;
   }
};
class part : public employee
{
```

```
public:
   void sal()
   {
      cout << "\nSalary of Part-time employee : " << pph * hr * 30 << endl;
   }
};
int main()
{
   regular r;
   employee *er = &r;
   er->info();
   er->regular();
   er->sal();
   part p;
   employee *ep = &p;
   ep->info();
   ep->part();
   ep->sal();
   return 0;
}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\11_11_2021> ./Employee
Enter name : Kashi
Enter ID : 109
Enter salary : 50000
Enter DA : 120
Enter HRA : 300

Salary of regular employee : 50420
Enter name : Rohan
Enter ID : 187
Enter number of hours : 8
Enter pay per hour : 100

Salary of Part-time employee : 24000
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\11_11_2021>
```

Question 3)Create a class which stores account number, customer name and balance. Derive two classes from 'Account' class: 'Savings' and 'Current'. The 'Savings' class stores minimum balance. The 'Current' class stores the over-due amount. Include member functions in the appropriate class for
-deposit money
-withdraw [For saving account minimum balance should be checked.]
[For current account overdue amount should be calculated.]
-display balance
Display data from each class using virtual function

```
#include <iostream>
using namespace std;
class account
{
public:
   int acn, balance, minbal, wd, dp, bal;
   char name[25];
   void info()
   {
      cout << "Enter account number : ";
      cin >> acn;
      cout << "Enter name : ";
      cin >> name;
      cout << "Enter balance : ";
      cin >> balance;
      cout << "Enter amount to withdraw : ";
      cin >> wd;
      cout << "Enter amount to deposit : ";
      cin >> dp;
   }
   void saving()
   {
      minbal = 1000;
      bal = balance - wd + dp;
      cout << "Minimum balance is : " << minbal << endl;
   }
   void current()
   {
      bal = balance - wd + dp;
      cout << "Current balance is : " << bal << endl;
   }
   virtual void data() = 0;
};
class savings : public account
{
public:
   void data()
   {
      cout << "Account number : " << acn << endl;
      cout << "Customer name : " << name << endl;
      if (bal < minbal)
         cout << "You cannow withdraw below minimum balance, which is Rs. " << minbal << endl;
```

```
      else
         cout << "Balance is : " << bal << endl;
   }
};
class current : public account
{
public:
   void data()
   {
      cout << "Account number : " << acn << endl;
      cout << "Customer name : " << name << endl;
      if (bal < 0)
      {
         cout << "Amount Overdued." << endl;
      }
      else
      {
         cout << "Balance is : " << bal << endl;
      }
   }
};
int main()
{
   int ch;
   savings s;
   account *as = &s;
   current c;
   account *ac = &c;
   while (1)
   {
      cout << "1. Savings" << endl;
      cout << "2. Current" << endl;
      cout << "3. Exit" << endl;
      cout << "Enter choice : ";
      cin >> ch;
      switch (ch)
      {
      case 1:
         cout << "Savings Account." << endl;
         as->info();
         as->saving();
         as->data();
         break;
      case 2:
         cout << "Current Account." << endl;
         ac->info();
         ac->current();
         ac->data();
         break;
      case 3:
         return 0;
         break;
      default:
         cout << "Wrong Choice!!" << endl;
         break;
      }
   }
}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\11_11_2021> g++ Account.cpp -oAccount
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\11_11_2021> ./Account
1. Savings
2. Current
3. Exit
Enter choice : 2
Current Account.
Enter account number : 123
Enter name : Someone
Enter balance : 20000
Enter amount to withdraw : 500
Enter amount to deposit : 4500
Current balance is : 24000
Account number : 123
Customer name : Someone
Balance is : 24000
1. Savings
2. Current
3. Exit
Enter choice : 3
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\11_11_2021>
```

Question 4) WAP to demonstrate use of pure virtual function and abstract base class.

```
#include <iostream>
using namespace std;
class Shape
{
protected:
   float dimension;

public:
   void Dimension()
   {
      cin >> dimension;
   }
   virtual float calculateArea() = 0;
};
class Square : public Shape
{
public:
   float calculateArea()
   {
      return dimension * dimension;
   }
};
class Circle : public Shape
{
public:
   float calculateArea()
   {
      return 3.14 * dimension * dimension;
   }
};
int main()
{
   Square square;
   Circle circle;
   cout << "enter the side of the square : ";
   square.Dimension();
   cout << "\narea of square:" << square.calculateArea() << endl;
   cout << "enter the radius of the circle : ";
   circle.Dimension();
   cout << "\narea of circle:" << circle.calculateArea() << endl;
   return 0;
}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\11_11_2021> ./pureVirtual
enter the side of the square : 4

area of square:16
enter the radius of the circle : 2

area of circle:12.56
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\11_11_2021>
```