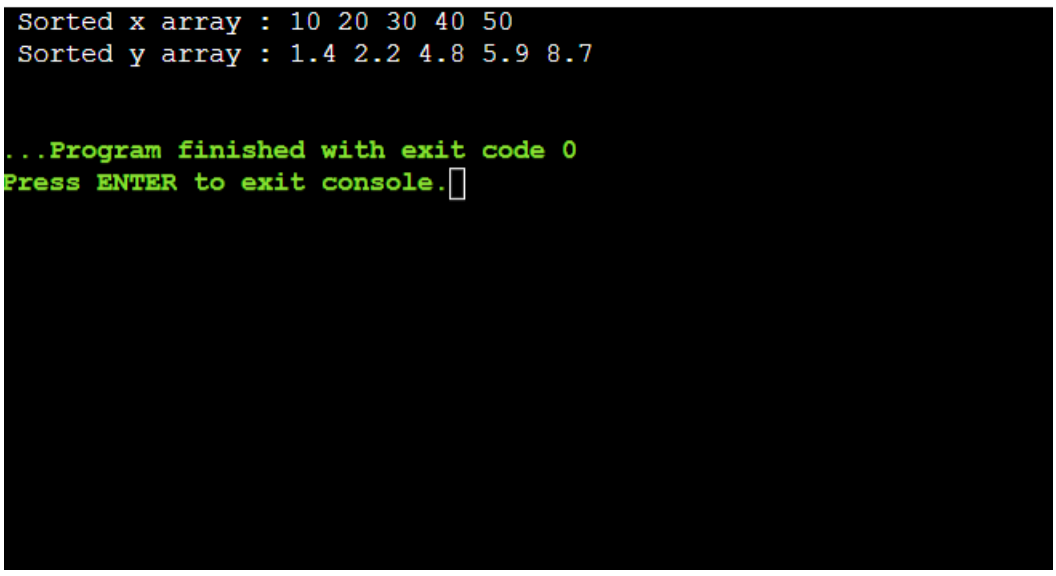
Experiment – 10 A

Code : -

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

template <class T>
void quickSort(T a[], int left, int right) {
   if (left >= right) return;

   int i = left, j = right;
   T pivot = a[(left + right) / 2];

   while (i <= j) {
      while (a[i] < pivot) i++;
      while (a[j] > pivot) j--;
      if (i <= j) {
         swap(a[i], a[j]);
         i++;
         j--;
      }
   }
   quickSort(a, left, j);
   quickSort(a, i, right);
}

template<class x>
void swap(x &a, x &b) {
   x temp = a;
   a = b;
   b = temp;
}
```

```
int main() {
    int x[5] = { 10, 50, 30, 40, 20 };
    float y[5] = {1.4, 2.2, 8.7, 4.8, 5.9};

    quickSort<int>(x, 0, 4);
    quickSort<float>(y, 0, 4);

    cout << " Sorted x array : ";
    for (int i = 0; i < 5; i++)
        cout << x[i] << " ";
    cout << endl;

    cout << " Sorted y array : ";
    for (int j = 0; j < 5; j++)
        cout << y[j] << " ";
    cout << endl;

    return 0;
}
```

Output : -



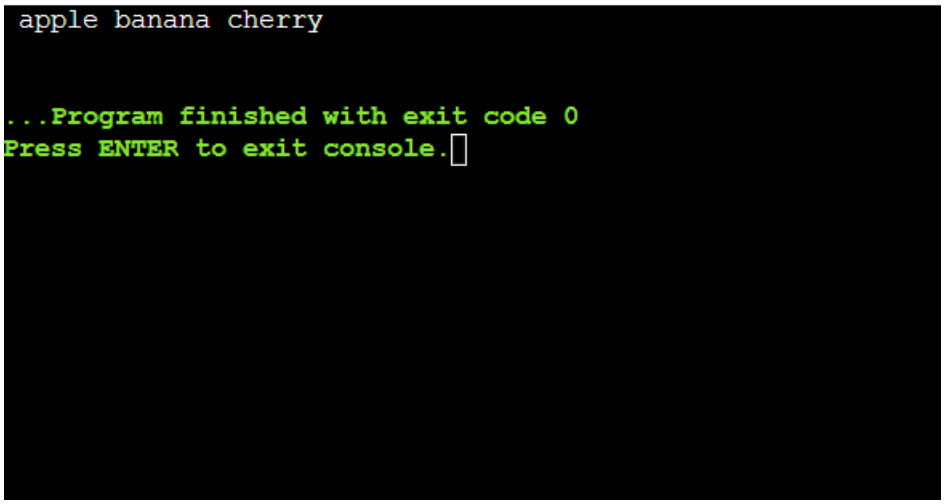
Experiment – 10 B

Code : -

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

template <typename T> class Array {
private:
   T* ptr;
   int size;

public:
   Array(T arr[], int s);
   void print();
};

template <typename T> Array<T>::Array(T arr[], int s)
{
   ptr = new T[s];
   size = s;
   for (int i = 0; i < size; i++)
      ptr[i] = arr[i];
}

template <typename T> void Array<T>::print()
{
   for (int i = 0; i < size; i++)
      cout << " " << *(ptr + i);
   cout << endl;
}
```

```
int main()
{
    string arr[3] = { "apple", "banana", "cherry" };
    Array<string> a(arr, 3);
    a.print();
    return 0;
}
```

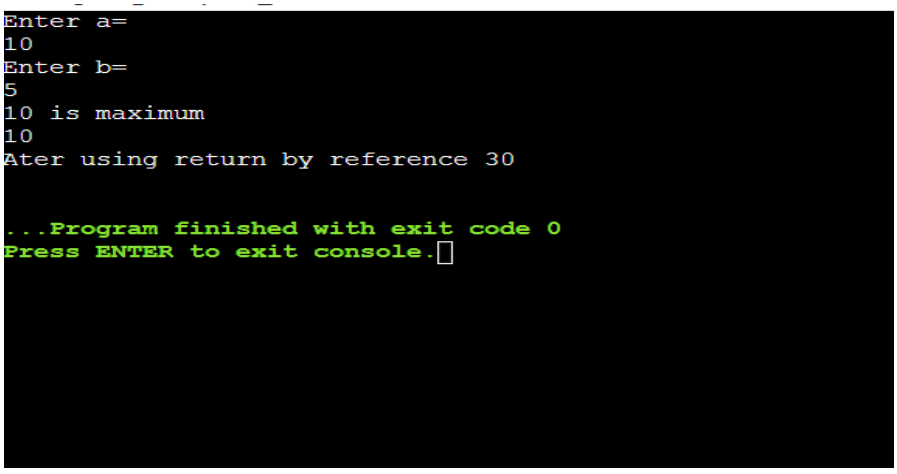
Output : -



Code : -

```
#include <iostream>
using namespace std;
int &max( int &a, int &b){
if (a>b){
cout<<a<<" is maximum"<<endl;
return a; }
else{
cout<<b<<" is maximum"<<endl;
return b; }}
int main(){
int a,b;
cout<<"Enter a="<<endl;
cin>>a;
cout<<"Enter b="<<endl;
cin>>b;
int &c=max(a,b);
cout<<c<<endl;
c=30;
cout<<"Ater using return by reference "<<c<<endl;
}
```

Output: -



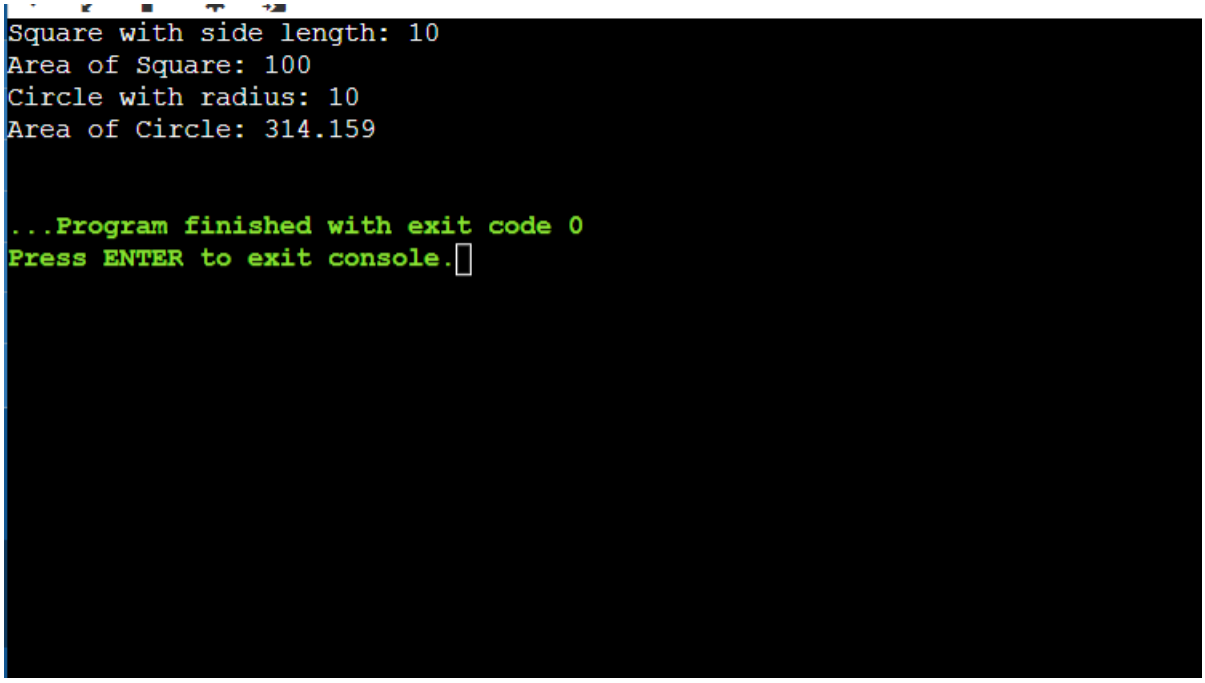
Experiment – 7 A

Code : -

```
#include <iostream>
using namespace std;
class Square {
   float side;
public:
   Square(float s) {
      side = s;
   }
   void display() {
      cout << “Square with side length: “ << side << endl;
      cout << “Area of Square: “ << side * side << endl;
   }
};
class Circle {
   float radius,area;
public:
   Circle(float r) {
      radius = r;
      area=3.14159 * radius * radius;
   }
   void show() {
      cout << “Circle with radius: “ << radius << endl;
      cout << “Area of Circle: “ << area << endl;
   }
   operator Square() {
      Square temp(radius );
      return temp;
   }
};
```

```
int main() {
    Circle c(10);
    Square s = c;
    s.display();
    c.show();
    return 0;
}
```

Output: -

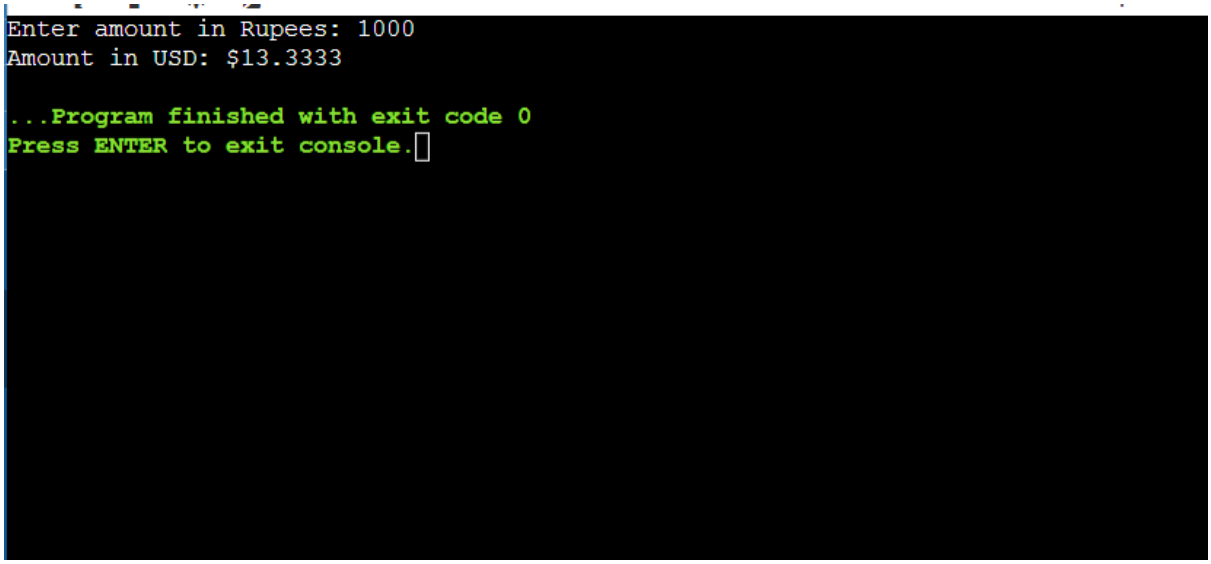


Experiment – 7 B

Code : -

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

class Currency {
   float amount;
public:
   Currency(){
      amount = 0;
   }
   Currency(float amt){
      amount = amt/75;
   }
   void showAmount(){
      cout << “Amount in USD: $” << amount;
   }
};
int main() {
   Currency curr1;
   float rupees;
   cout << “Enter amount in Rupees: “;
   cin >> rupees;
   curr1 = rupees;
   curr1.showAmount();

   return 0;
}
```

Output:-


Enter amount in Rupees: 1000
Amount in USD: $13.3333
...Program finished with exit code 0
Press ENTER to exit console.

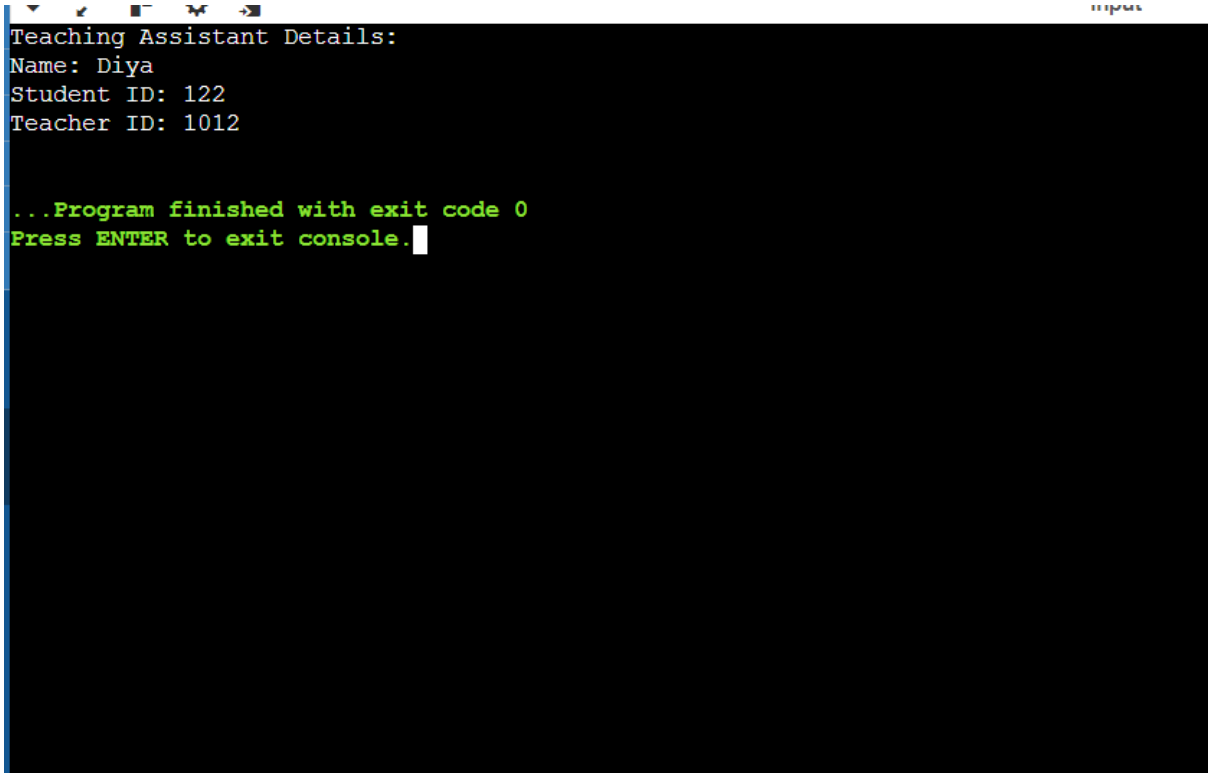
Experiment – 7 C1

Code :-

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

class Person {
public:
   string name;

   Person(string n) : name(n) {}

   void displayPerson() {
      cout << “Name: “ << name << endl;
   }
};
class Student : virtual public Person {
public:
   int studentID;

   Student(string n, int id) : Person(n), studentID(id) {}

   void displayStudent() {
      cout << “Student ID: “ << studentID << endl;
   }
};
class Teacher : virtual public Person {
public:
   int teacherID;

   Teacher(string n, int id) : Person(n), teacherID(id) {}
```

```
    void displayTeacher() {
        cout << “Teacher ID: “ << teacherID << endl;
    }
};

class TA : public Student, public Teacher {
public:
    TA(string n, int studentID, int teacherID)
        : Person(n), Student(n, studentID), Teacher(n, teacherID) {}

    void displayTA() {
        cout << “Teaching Assistant Details:” << endl;
        displayPerson();
        displayStudent();
        displayTeacher();
    }
};

int main() {
    TA ta(“Diya”, 122, 1012);

    ta.displayTA();

    return 0;
}
```

Output: -


Teaching Assistant Details:
Name: Diya
Student ID: 122
Teacher ID: 1012
...Program finished with exit code 0
Press ENTER to exit console.

Experiment – 7 C2

Code : -

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

class Square {
   int side;
public:
   Square(int s) : side(s) {}
   int getSide() { return side; }
   void display() {
      cout << "Side: " << side << "\nArea of Square: " << side * side << endl;
   }
};
class Circle {
   float radius;
public:
   Circle(Square s) {
      radius = s.getSide();
   }

   float getRadius() { return radius; }

   void show() {
      cout << "Radius: " << radius << "\nArea of Circle: " << 3.14159 * radius * radius << endl;
   }
};
int main() {
   Square s(10);
   Circle c = s;
   c.show();
```

```
    s.display();


    return 0;
}
```

Output:-

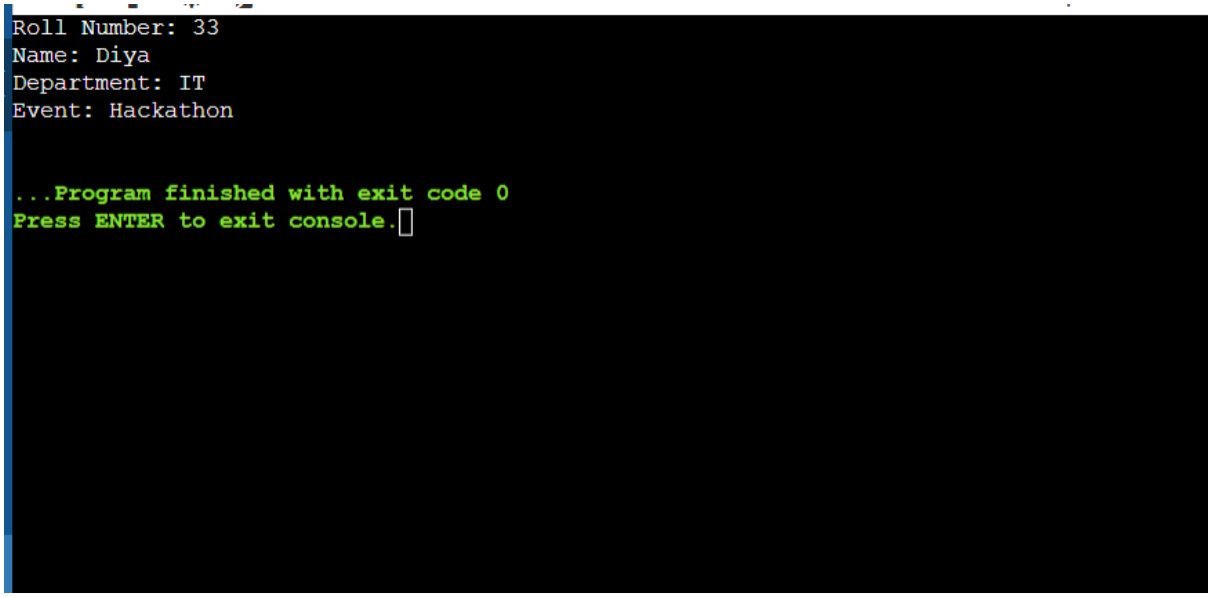
Experiment - 8

Code : -

```
#include<iostream>
using namespace std;
class Student {
   public:
      int rollno;
      string name;
      Student(int r, string n) {
         rollno = r;
         name = n;
      }
      void print() {
         cout << "Roll Number: " << rollno << endl;
         cout << "Name: " << name << endl;
      }
};
class GenSecretary : public Student {
   public:
      string dept;
      string event;
      GenSecretary(int r, string n, string d, string e) : Student(r, n) {
         dept = d;
         event = e;
      }
      void show() {
         cout << "Department: " << dept << endl;
         cout << "Event: " << event << endl;
      }
};
```

```
int main() {
    GenSecretary gs(33, "Diya", "IT", "Hackathon");

    gs.print();
    gs.show();

    return 0;
}
```

Output : -



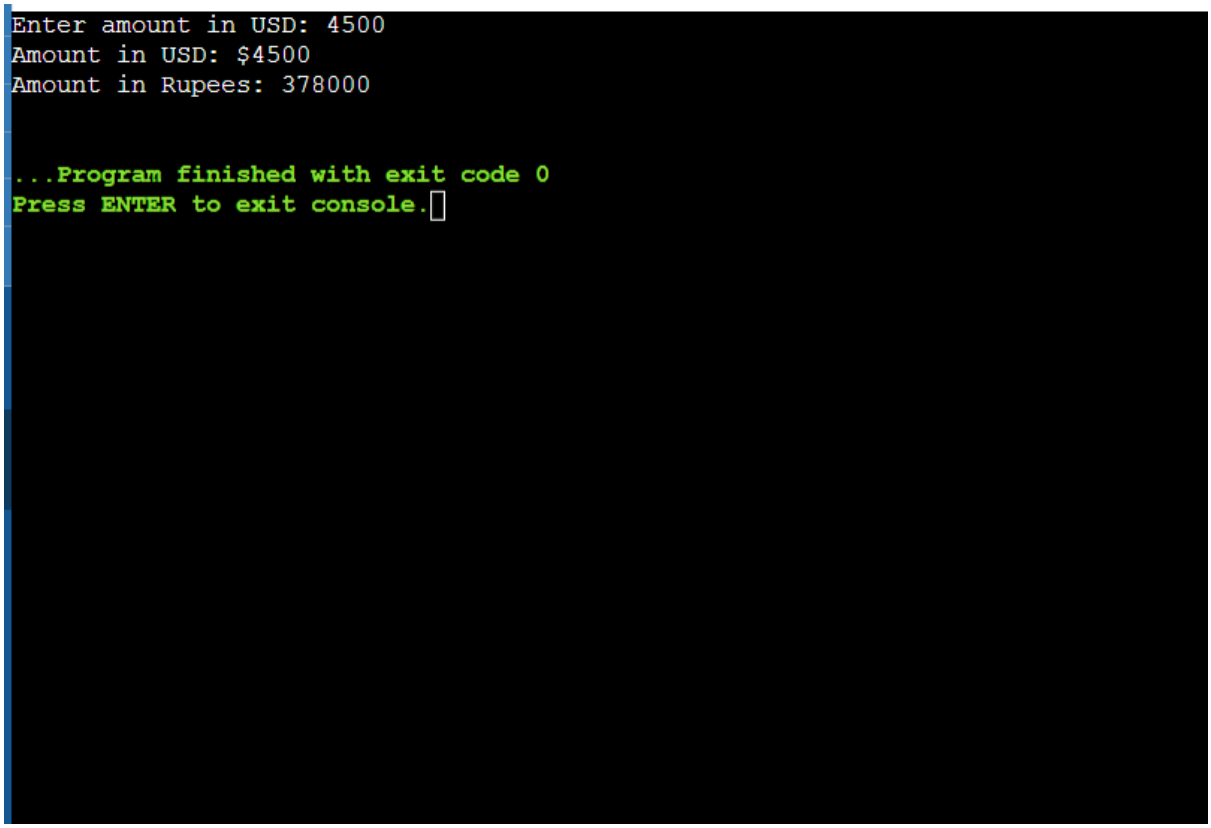
Experiment - 9

Code : -

```
#include <iostream>
using namespace std;
class Currency {
   float amount;
public:
   Currency() {
      amount = 0;
      }
   operator float() {
      return amount * 84;
   }
   void getAmount() {
      cout << "Enter amount in USD: ";
      cin >> amount;
   }
   void showAmount() {
      cout << "Amount in USD: $" << amount << endl;
   }
};
int main() {
   Currency curr1;
   float rupees;
   curr1.getAmount();
   rupees = curr1;
   curr1.showAmount();
   cout << "Amount in Rupees: " << rupees << endl;
   return 0;
```

}

Output : -


Enter amount in USD: 4500
Amount in USD: $4500
Amount in Rupees: 378000
...Program finished with exit code 0
Press ENTER to exit console.