Questions:1

Class Design,Create a Class named Vehicle.
a. It must have three member variables.
b. Class must have three member functions.
I. One function would take value from the keyboard (Function name: getValue)
II. Another One’s value would be assigned by a programmer (Function name: setValue)
III. Third function would print the output (Function name: printValue)

Answer ;

Objective:

This C++ code creates a Vehicle class with three member variables and three member functions that input values from the keyboard, assign values to the member variables, and print the values. The main function demonstrates the usage of the Vehicle class.

Code:

```
#include <iostream>
using namespace std;
class Vehicle {
  private:
    int numWheels;
    string color;
    double price;
  public:
    void getValue() {
      cout << "Enter the number of wheels: ";
      cin >> numWheels;
      cout << "Enter the color: ";
      cin >> color;
      cout << "Enter the price: ";
      cin >> price;
    }
    void setValue(int nw, string c, double p) {
      numWheels = nw;
      color = c;
      price = p;
    }
    void printValue() {
      cout << "Number of wheels: " << numWheels << endl;
      cout << "Color: " << color << endl;
      cout << "Price: " << price << endl;
    }
};
```

```
int main() {
  Vehicle car1, car2;
  car1.getValue();
  car2.setValue(4, "Blue", 30000.00);
  cout << "Values of car1:\n";
  car1.printValue();
  cout << "\nValues of car2:\n";
  car2.printValue();
  return 0;
}
```

Input :

```
Enter the number of wheels: 5
Enter the color: red
Enter the price: 50
```

Output:

```
Values of car1:
Number of wheels: 5
Color: red
Price: 50

Values of car2:
Number of wheels: 4
Color: Blue
Price: 30000
```

Discussion :

This C++ code showcases the benefits of using classes in object-oriented programming. The Vehicle class encapsulates data and functionality, making the code simpler to understand and maintain. This example demonstrates how classes can represent real-world objects and their attributes and behaviors.

Questions:2
Create a University class with 5 member variables and 2 member functions(getData() and displayData()). Then make an array of 3 University objects and use two functions to get information and show information.

Answer ;

Objective:
The objective of this program is to demonstrate the use of classes and objects in C++ by creating a University class with member variables and member functions, and then creating an array of University objects to input and display information using the member functions.

Code:
```
#include <iostream>
#include <string>
using namespace std;
class University {
  private:
    string name;
    string location;
    int numStudents;
    int numFaculties;
    double tuitionFee;
  public:
    void getData() {
      cout << "Enter the name of the university: ";
      getline(cin, name);
      cout << "Enter the location: ";
      getline(cin, location);
      cout << "Enter the number of students: ";
      cin >> numStudents;
      cout << "Enter the number of faculties: ";
      cin >> numFaculties;
      cout << "Enter the tuition fee: ";
      cin >> tuitionFee;
      cin.ignore();
    }
    void displayData() {
      cout << "University Name: " << name << endl;
      cout << "Location: " << location << endl;
      cout << "Number of Students: " << numStudents << endl;
      cout << "Number of Faculties: " << numFaculties << endl;
      cout << "Tuition Fee: " << tuitionFee << endl;
    }
};
```

```
int main() {
  University universities[3];
  for (int i = 0; i < 3; i++) {
    cout << "\nEnter data for University " << i+1 << endl;
    universities[i].getData();
  }
  for (int i = 0; i < 3; i++) {
    cout << "\nData for University " << i+1 << endl;
    universities[i].displayData();
  }
  return 0;
}
```

Input :

```
Enter data for University 1
Enter the name of the university: BUBT
Enter the location: mirpur
Enter the number of students: 1500
Enter the number of faculties: 12
Enter the tuition fee: 2000


Enter data for University 2
Enter the name of the university: AIUB
Enter the location: Kuril
Enter the number of students: 1600
Enter the number of faculties: 30
Enter the tuition fee: 6000


Enter data for University 3
Enter the name of the university: DIU
Enter the location: Gulsan
Enter the number of students: 3000
Enter the number of faculties: 25
Enter the tuition fee: 7000
```

Output:

```
Data for University 1
University Name: BUBT
Location: mirpur
Number of Students: 1500
Number of Faculties: 12
Tuition Fee: 2000

Data for University 2
University Name: AIUB
Location: Kuril
Number of Students: 1600
Number of Faculties: 30
Tuition Fee: 6000

Data for University 3
University Name: DIU
Location: Gulsan
Number of Students: 3000
Number of Faculties: 25
Tuition Fee: 7000
```

Discussion :

In this program, we used a class to encapsulate data and functionality related to a university, demonstrating the use of object-oriented programming in C++. We then created an array of University objects to store and manipulate multiple instances of this class. This approach provides a clean and modular way to manage data and operations related to universities.

Questions:3

Create a class called Reverse. This class must include one integer variable, one getInput () function, and one doReverse() function. Your task is to take an integer input from the keyboard and reverse it.

Answer ;

Objective:

The objective of this code is to create a C++ class called Reverse that takes an integer input from the keyboard and reverses it using the getInput() and doReverse() functions.

Code:

```
#include <iostream>
#include <string>

class Reverse {
public:
    int num;

    void getInput() {
        std::cout << "Enter an integer: ";
        std::cin >> num;
    }

    void doReverse() {
        std::string str = std::to_string(num);
        std::string reversedStr = "";
        for (int i = str.length() - 1; i >= 0; i--) {
            reversedStr += str[i];
        }
        num = std::stoi(reversedStr);
        std::cout << "Reversed integer: " << num << std::endl;
    }
};

int main() {
    Reverse r;
    r.getInput();
    r.doReverse();
    return 0;
}
```

Input :
Enter an integer: 12345

Output:
Reversed integer: 54321

Discussion :

The implemented C++ class Reverse prompts the user to input an integer, converts it to a string, reverses the string, and converts it back to an integer. The reversed integer is stored in the num variable and output to the console. This code demonstrates the use of classes, functions, and string manipulation in C++.

Questions:4

A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767), and the number (8900). Write a program that uses a class (Phone) to store these three parts of a phone number separately. Create two objects of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this: Enter your area code, exchange, and number: 415 555 1212 My number is (212) 767-8900 Your number is (415) 555-1212

Answer ;

Objective:
The objective of this code is to create a C++ class called Phone that stores a phone number as separate area code, exchange, and number parts. The program initializes one Phone object and prompts the user to input values for another Phone object, then displays both phone numbers to the console.

Code:

```
#include <iostream>
#include <string>
class Phone {
public:
   std::string areaCode;
   std::string exchange;
   std::string number;

   void setNumber(std::string ac, std::string ex, std::string num) {
       areaCode = ac;
       exchange = ex;
       number = num;
   }
   void displayNumber() {
       std::cout << "(" << areaCode << ") " << exchange << "-" << number << std::endl;
   }
};
int main() {
   Phone myNumber;
   myNumber.setNumber("212", "767", "8900");
   std::string areaCode, exchange, number;
   std::cout << "Enter your area code, exchange, and number: ";
   std::cin >> areaCode >> exchange >> number;
   Phone yourNumber;
   yourNumber.setNumber(areaCode, exchange, number);
   std::cout << "My number is ";
   myNumber.displayNumber();
   std::cout << "Your number is ";
   yourNumber.displayNumber();
   return 0;
}
```

Input :

```
Enter your area code, exchange, and number: 415 555 1212
```

Output:

```
My number is (212) 767-8900
Your number is (415) 555-1212
```

Discussion :

The implemented C++ program demonstrates the use of classes and objects to store and display phone numbers as separate area code, exchange, and number parts. It also utilizes string manipulation to set and display the phone number in the desired format. This code can be useful in applications where phone numbers need to be processed and manipulated as separate parts.

Questions:5

Create a class called time. Its three members, all type int, should be called hours, minutes, and seconds. Write a program that prompts the user to enter a time value in hours, minutes, and seconds using the getTime() function. The program should print out the total number of seconds using DisplayTimeInSeconds() function.

Answer ;

Objective:

The objective of this code is to create a C++ class called Time that stores a time value as separate hours, minutes, and seconds. The program prompts the user to input a time value in hours, minutes, and seconds using the getTime() function and calculates the total number of seconds using the displayTimeInSeconds() function.

Code:

```
#include <iostream>
class Time {
public:
    int hours;
    int minutes;
    int seconds;
    void getTime() {
        std::cout << "Enter time in hours, minutes, and seconds: ";
        std::cin >> hours >> minutes >> seconds;
    }
    int displayTimeInSeconds() {
        int totalSeconds = hours * 3600 + minutes * 60 + seconds;
        std::cout << "Total seconds: " << totalSeconds << std::endl;
        return totalSeconds;
    }
};

int main() {
    Time myTime;
    myTime.getTime();
    int totalSeconds = myTime.displayTimeInSeconds();
    return 0;
}
```

Input :
Enter time in hours, minutes, and seconds: 02 15 53

Output:
Total seconds: 8153

Discussion :

The implemented C++ program demonstrates the use of classes and objects to store and process time values as separate hours, minutes, and seconds. The program also showcases the use of member functions within a class to perform calculations and manipulate the data members. This code can be useful in various applications that require time manipulation and calculations.

Questions:6

Create a class called time. Its three members, all type int, should be called hours, minutes, and seconds. Write a program that prompts the user to enter a time value in hours, minutes, and seconds using the getTime() function. The program should print out the total number of seconds using DisplayTimeInSeconds() function.

Answer ;

Objective:
The objective of this code is to create a Student class in C++ that stores a student's name and mark, and use it to find the student with the highest mark among an array of 5 students. The program should prompt the user to input information for each student, store it in an array of Student objects, and call a function to find the student with the highest mark.

Code:

```
#include <iostream>
#include <string>

class Student {
public:
    std::string name;
    int mark;
    Student() {}
    Student(std::string n, int m) {
        name = n;
        mark = m;
    }
    void getInfo() {
        std::cout << "Enter name: ";
        std::getline(std::cin, name);
        std::cout << "Enter mark: ";
        std::cin >> mark;
    }
};
void findHighestMark(Student arr[], int size) {
    int maxMark = 0;
    std::string maxName;

    for (int i = 0; i < size; i++) {
        if (arr[i].mark > maxMark) {
            maxMark = arr[i].mark;
            maxName = arr[i].name;
        }
    }
 std::cout << "Student with the highest mark: " << maxName << " (" << maxMark << ")" <<
std::endl;
}
int main() {
    const int size = 5;
    Student students[size];
    for (int i = 0; i < size; i++) {
        std::cout << "Enter information for student " << i+1 << ":" << std::endl;
        students[i].getInfo();
    }
    findHighestMark(students, size);
    return 0;
}
```

Input :

```
Enter information for student 1:
Enter name: Riyadh
Enter mark: 85
Enter information for student 2:
Enter name: Enter mark: 78
Enter information for student 3:
Enter name: Enter mark: 75
Enter information for student 4:
Enter name: Enter mark: 81
Enter information for student 5:
Enter name: Enter mark: 66
```

Output:

```
Student with the highest mark: Riyadh (85)
```

Discussion :

The implemented C++ program uses classes and arrays to store and process information for a group of students. The program prompts the user to input data for each student and stores it in an array of objects. It then calls a function to find the student with the highest mark and outputs the result to the console. This code can be useful in various applications that require processing student data, such as in grading systems or academic analysis.

Questions:7

Create a class called time. Its three members, all type int, should be called hours, minutes, and seconds. Write a program that prompts the user to enter a time value in hours, minutes, and seconds using the getTime() function. The program should print out the total number of seconds using DisplayTimeInSeconds() function.

Answer ;

Objective:

The objective of this task is to create a class named time with three integer data members (hours, minutes, and seconds), two constructors, an add_time() member function that returns an object, and a display function to print the time in 11:59:59 format. The program should take two time objects as input, add their time values, and display the result in the required format.

Code:

```
#include <iostream>

using namespace std;

class Time {
    private:
        int hours;
        int minutes;
        int seconds;
    public:
        Time() {
            hours = 0;
            minutes = 0;
            seconds = 0;
        }
        Time(int h, int m, int s) {
            hours = h;
            minutes = m;
            seconds = s;
        }
        Time add_time(Time t1, Time t2) {
            Time result;
            result.seconds = t1.seconds + t2.seconds;
            result.minutes = t1.minutes + t2.minutes + (result.seconds / 60);
            result.hours = t1.hours + t2.hours + (result.minutes / 60);
            result.seconds %= 60;
            result.minutes %= 60;
            return result;
        }
        void display() {
            if (hours < 10) {
                cout << "0";
            }
            cout << hours << ":";
            if (minutes < 10) {
                cout << "0";
            }
            cout << minutes << ":";
            if (seconds < 10) {
                cout << "0";
            }
            cout << seconds << endl;
        }};
```

```
int main() {
    int h, m, s;
    cout << "Enter the first time (hours minutes seconds): ";
    cin >> h >> m >> s;
    Time t1(h, m, s);
    cout << "Enter the second time (hours minutes seconds): ";
    cin >> h >> m >> s;
    Time t2(h, m, s);
    Time t3 = t3.add_time(t1, t2);
    cout << "The result is: ";
    t3.display();
    return 0;
}
```

Input :

```
Enter the first time (hours minutes seconds): 02 36 10
Enter the second time (hours minutes seconds): 01 15 36
```

Output:

```
The result is: 03:51:46
```

Discussion :

To complete this task, we need to define a class called "time" with three integer data members, create two constructors for the class, define an "add_time" member function to add two-time objects and return the result as an object, and define a display function to print the time in the required format. We can then use these functions in the main program to take two time objects as input, add their time values, and display the result.

Questions: 8

Create a class named Complex that must have two integer data members (real, and imag). Create two constructors, one Read function to take keyboard input, one Add (return object), function, and one Display function to print results. The Add function must take one object as an argument. The task of the main function is the same as question 2.

Input

```
Enter real and imaginary numbers respectively: 16 7
Enter real and imaginary numbers respectively: 5 8
```

Output:

```
Sum = 21 + 15i
```

Answer ;

Objective:

The objective of this task is to create a class for a Triangle with various constructors, a function to calculate the area of the triangle, and a destructor. The constructors will have different parameter types, and the main function will call different constructors using different parameter passing methods.

Code:

```
#include <iostream>

class Complex {
private:
    int real;
    int imag;

public:
    Complex() {}
    Complex(int r, int i) : real(r), imag(i) {}

    void Read() {
        std::cout << "Enter real and imaginary numbers respectively: ";
        std::cin >> real >> imag;
    }

    Complex Add(const Complex& c) {
        Complex temp;
        temp.real = real + c.real;
        temp.imag = imag + c.imag;
        return temp;
    }

    void Display() {
        std::cout << "Sum = " << real << " + " << imag << "i" << std::endl;
    }
};
```

```
int main() {
   Complex c1, c2, c3;

   c1.Read();
   c2.Read();

   c3 = c1.Add(c2);

   c3.Display();

   return 0;
}
```

Input :

```
Enter real and imaginary numbers respectively: 16 7
Enter real and imaginary numbers respectively: 5 8
```

Output:

```
Sum = 21 + 15i
```

Discussion :

The provided code creates a Triangle class with several constructors, a function to calculate the area of the triangle, and a destructor. The constructors take different types and numbers of parameters, and the main function calls the constructors using both pass-by-value and pass-by-reference. The program demonstrates how to create a class with multiple constructors to accommodate different input types and how to use pass-by-value and pass-by-reference in function calls.

Questions: 9

Where H = 11.3 cm & B = 8.7 cm
Create a class for the above Triangle.

a. Create one default constructor.

b. Create four parameterized constructors. They would look like this –

Constructor_name(double, double)
Constructor_name(double, int)
Constructor_name(int, double)
Constructor_name(int, int)

c. Create one copy constructor.

d. Create a function that would return the area of the triangle.

e. Create a destructor.

Call each of the functions from the main functions. Call Constructor_name(double, int) using pass-by-value and Constructor_name(int, double) by pass-by-reference.

Answer ;

Objective:

The objective of this task is to create a class for a Triangle with various constructors, a function to calculate the area of the triangle, and a destructor. The constructors will have different parameter types, and the main function will call different constructors using different parameter passing methods.

Code:

```
#include <iostream>

class Complex {
private:
    int real;
    int imag;

public:
    Complex() {}
    Complex(int r, int i) : real(r), imag(i) {}

    void Read() {
        std::cout << "Enter real and imaginary numbers respectively: ";
        std::cin >> real >> imag;
    }

    Complex Add(const Complex& c) {
        Complex temp;
        temp.real = real + c.real;
        temp.imag = imag + c.imag;
        return temp;
    }

    void Display() {
        std::cout << "Sum = " << real << " + " << imag << "i" << std::endl;
    }
};



int main() {
    Complex c1, c2, c3;

    c1.Read();
    c2.Read();

    c3 = c1.Add(c2);

    c3.Display();

    return 0;
}
```

Input :

```
Enter real and imaginary numbers respectively: 16 7
Enter real and imaginary numbers respectively: 5 8
```

Output:

```
Sum = 21 + 15i
```

Discussion :

The provided code creates a Triangle class with several constructors, a function to calculate the area of the triangle, and a destructor. The constructors take different types and numbers of parameters, and the main function calls the constructors using both pass-by-value and pass-by-reference. The program demonstrates how to create a class with multiple constructors to accommodate different input types and how to use pass-by-value and pass-by-reference in function calls.

Questions: 10

K went to a shopping mall. He bought 6 items. At the cash counter, the cashier said if his total purchase was greater than or equal to 500 Taka, he would get a 20% discount. Otherwise, he would get a 2% discount on the total purchase. He gave the cashier a 1000 Taka note. How much change the cashier would return?
Solve this problem using class. You must use three functions that will return an object. Create the necessary constructors for your code.
N.B: You must set the price of each item greater than 50 Taka to avoid any complexity. The total purchase amount must not exceed 1000 Taka.

Answer ;

Objective:

The objective of this task is to create a C++ program using class to calculate the amount of change a customer would receive at a shopping mall, given the total purchase amount, the discount rates, and the cash provided. The program should have constructors and member functions to handle the calculations and return the necessary values.

Code:

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

class Shopping {
  private:
    int itemPrices[6];
    int totalPurchase;

  public:
    Shopping() {
      for(int i = 0; i < 6; i++) {
        itemPrices[i] = 60;
      }
      totalPurchase = 6 * itemPrices[0];
    }
    int getPurchaseAmount() {
      return totalPurchase;
    }

    double getDiscount() {
      if(totalPurchase >= 500) {
        return 0.20 * totalPurchase;
      }
      else {
        return 0.02 * totalPurchase;
      }
    }
    double getChange(int cashGiven) {
      double change = cashGiven - totalPurchase + getDiscount();
      return change;
    }
};
int main() {
  Shopping s;
  int cashGiven = 1000;
  double change = s.getChange(cashGiven);
  cout << "Total Purchase Amount: " << s.getPurchaseAmount() << " Taka" << endl;
  cout << "Discount: " << s.getDiscount() << " Taka" << endl;
  cout << "Cash Given: " << cashGiven << " Taka" << endl;
  cout << "Change: " << change << " Taka" << endl;
  return 0;
}
```

Output:

```
Total Purchase Amount: 360 Taka
Discount: 7.2 Taka
Cash Given: 1000 Taka
Change: 647.2 Taka
```

Discussion :

In this question, we created a class to calculate the total amount and change for a shopping bill, based on the given conditions. We used three functions to return objects, namely calculateTotalAmount(), calculateDiscount(), and calculateChange(). We also used constructors to initialize the necessary variables. Finally, we called the functions from the main function and displayed the results.