

Chapter 2. C++

Q1.

Q1-1. Read the following C program and complete it by filling the blanks.

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

double income_tax_rate(_____ );

void main()
{
    int j;
    _____ income;
    double tax_rate;

    for (j=1; j<=3_____ j++)
    {
        income=30000.0+10000.0*j;
        tax_rate=income_tax_rate(income);
        printf("Income tax for income of %5.2f is: %1.2f . \n",
               _____, income, tax_rate);
    }

    _____ income_tax_rate(double Inc)
    {
        double _____;
        if (Inc<=30000.0) { Rate=0.16; };
        if ((Inc>30000.0) && (Inc<=50000.0)) { Rate=0.18; };
        if (Inc>50000.0) { Rate=0.21; };
        return(Rate);
    }
}
```

Q1-2. Draw a structure diagram for the above C program.

Q1-3. Write the expected **print result** generated by the above C program.

Answers:

Q1-1 Answer:

```

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

double income_tax_rate(double);

void main()
{
    int j;
    double income;
    double tax_rate;

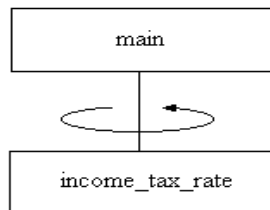
    for (j=1; j<=3; j++)
    {
        income=30000.0+10000.0*j;
        tax_rate=income_tax_rate(income);
        printf("Income tax for income of %5.2f is: %1.2f . \n",
               income, tax_rate);
    }
}

double income_tax_rate(double Inc)
{
    double Rate;
    if (Inc<=30000.0)           { Rate=0.16; };
    if ((Inc>30000.0) && (Inc<=50000.0)) { Rate=0.18; };
    if (Inc>50000.0)           { Rate=0.21; };
    return(Rate);
}

```

Q1-2. Draw a structure diagram for the above C program.

Answer:



Q1-3. Write the expected **print result** generated by the above C program.

Answer:

Income tax for income of 40000.0 is: 0.18 .
Income tax for income of 50000.0 is: 0.18 .
Income tax for income of 60000.0 is: 0.21 .

=====

Q2.

Q2-1. Read the following C program and complete it by filling the blanks.

```
#include<iostream>
#include<cstring>
```

```

using namespace std;

double Discuout_Table(____);

void main()
{
    int i;
    _____ order;
    _____ Discount_Rate;

    _____ (i=1; i<=3; i++)
    {
        order=100.0*i;
        Discount_Rate=Discount_Table(order);
        printf("The discount rate for order of %4.2f is : %2.2f . \n",
            order, Discount_Rate);
    }
}

double Discount_Table(_____ K)
{
    double _____;
    if (K<=99) { D_R=0.00; };
    if ((K>99) && (K<=199)) { D_R=0.20; };
    if ((K>199) && (K<=299)) _____ { D_R=0.35; };
    if (K>299) { D_R=0.40; };
    return(D_R);
}

```

Q2-2. Draw a structure diagram for the above C program.

Q2-3. Write the expected **print result** generated by the above C program.

Answers:

Q2-1 Answer:

```

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

double Discuout_Table(double);

void main()
{
    int i;
    double order;
    double Discount_Rate;

    for (i=1; i<=3; i++)
    {

```

```

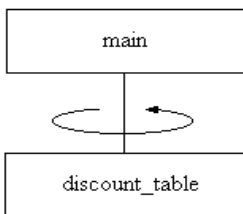
        order=100.0*i;
        Discount_Rate=Discount_Table(order);
        printf("The discount rate for order of %4.2f is : %2.2f . \n",
               order, Discount_Rate);
    }
}

double Discount_Table(double K)
{
    double D_R;
    if (K<=99)           { D_R=0.00; };
    if ((K>99) && (K<=199)) { D_R=0.20; };
    if ((K>199) && (K<=299)) { D_R=0.35; };
    if (K>299)           { D_R=0.40; };
    return(D_R);
}

```

Q2-2. Draw a structure diagram for the above C program.

Answer:



Q2-3. Write the expected **print result** generated by the above C program.

Answer:

The discount rate for order of 100.0 is : 0.20.
 The discount rate for order of 200.0 is : 0.35 .
 The discount rate for order of 300.0 is : 0.40 .

Q3.

Q3-1. Read the following C program and complete it by filling the blanks.

```

#_____<iostream>

using namespace std;

_____ insurance(_____);

void main()
{

```

```

int j;
int _____;
_____ monthly_rent;

for (j=1; j<4; j++)
{
    mileage=1000*j;
    monthly_rent= 0.2*mileage + insurance(mileage);
    printf("Monthly rent for %5d is : $ _____ . \n",
           mileage, monthly_rent);
}
}

```

```

_____ insurance(int miles)
{
    double mileage_charge;
    if (miles<=1000)           { mileage_charge=100.0; };
    if ((miles>1000) && (miles<=2000) _____
                                   { mileage_charge=150.0; };
    if (miles>2000)           { mileage_charge=200.0; };
    _____ (mileage_charge);
}

```

Q3-2. Draw a structure diagram for the above C program.

Q3-3. Write the expected **print result** generated by the above C program.

Answers:

Q3-1 Answer:

```

#include<iostream>
using namespace std;
double insurance(int);

void main()
{
    int j;
    int mileage;
    double monthly_rent;

    for (j=1; j<4; j++)
    {
        mileage=1000*j;
        monthly_rent= 0.2*mileage + insurance(mileage);
        printf("Monthly rent for %5d is : $ %4.2f . \n",
               mileage, monthly_rent);
    }
}

```

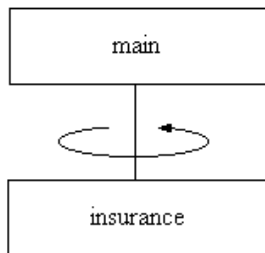
```
}

```

```
double insurance(int miles)
{
    double mileage_charge;
    if (miles<=1000)           { mileage_charge=100.0; };
    if ((miles>1000) && (miles<=2000))
                                { mileage_charge=150.0; };
    if (miles>2000)           { mileage_charge=200.0; };
    return(mileage_charge);
}
```

Q3-2. Draw a structure diagram for the above C program.

Answer:



Q3-3. Write the expected **print result** generated by the above C program.

Answer:

Monthly rent for 1000 is : \$ 300.0 .
 Monthly rent for 2000 is : \$ 550.0.
 Monthly rent for 3000 is : \$ 800.0 .

=====

Q4.

Q4-1. Read the following C++ program and complete it by filling the blanks.

```
#include<iostream>
```

```
#include<cstring>
```

```
_____ "student.h"
```

```
using namespace std;
```

```
// The header file of the class STUDENT used is included
```

```
_____()
```

```
{
```

```
    printf( "Student Name      Credits Needed for Graduation  \n");
```

```
    printf( "\n");
```

```

        for (int i=1; i<=3; i++) {
            STUDENT _____;
            Student.InitStudent(i);
            printf ("%s", Student.ReturnStudent_Name());
printf("%4d \n", Student.Credit_Due());
        }
    }

**** Class STUDENT definition ****
// File is "student.h"

class STUDENT

// Class declaration
{
    private:

// Attributes
    _____ Student_Number;
    _____ Student_Name[20];
    int Credit_Earned;
    char Student_Phone[15];

    public:
    _____ ()
    {
// Constructor is actually implemented by the method
// void InitStudent(int) depending on S_N
    };

// Operations

// The following procedure simulates the system to read a
// database/data file which records information of students

    _____ InitStudent(int S_N) {
        Student_Number=S_N;
        if (Student_Number==1)
            { _____(Student_Name, "John ");
              Credit_Earned=60;
              strcpy(Student_Phone, "123-1234 ");
            }
        if (Student_Number==2)
            { strcpy(Student_Name, "Anne ");
              Credit_Earned=123;
              strcpy(Student_Phone, "123-2345 ");
            }
        if (Student_Number==3)
            { strcpy(Student_Name, "Greg ");
              Credit_Earned=90;

```



```

        strcpy(Student_Phone, "123-7890 ");
    }

//  Next are methods of the STUDENT class ...

char *ReturnStudent_Name()
    {return Student_Name;};

_____ *ReturnStudent_Phone()
    {_____ Student_Phone;};

int ReturnCredit_Earned()
    {return Credit_Earned;};

int _____() {
    int Due_Amount;
    if ((Credit_Earned)<120)
        { Due_Amount=120-Credit_Earned; }

    _____
        { Due_Amount=0;  };

    _____(Due_Amount);
}
};

```

Q4-2. Draw an object-oriented analysis diagram for the above C++ program.

Q4-3. Write the expected **print result** generated by the above C++ program.

Answers:

Q4-1 Answer:

```

#include<iostream>
#include<cstring>

using namespace std;

#include"student.h"

// The header file of the class STUDENT used is included

void main()
{

    printf( "Student Name      Credits Needed for Graduation  \n");
    printf( "\n");

```

```

        for (int i=1; i<=3; i++) {

            STUDENT Student;
            Student.InitStudent(i);
            printf ("%s", Student.ReturnStudent_Name());
            printf ("%4d \n", Student.Credit_Due());
        }
    }

// **** Class STUDENT definition ****
// File is "student.h"

class STUDENT

// Class declaration
{
    private:

// Attributes
    int      Student_Number;
    char      Student_Name[20];
    int      Credit_Earned;
    char      Student_Phone[15];

    public:
        STUDENT ()
        {
// Constructor is actually implemented by the method
// void InitStudent(int) depending on S_N
        };

// Operations

// The following procedure simulates the system to read a
// database/data file which records information of students

    void InitStudent(int S_N) {
        Student_Number=S_N;
        if (Student_Number==1)
            { strcpy(Student_Name, "John");
              Credit_Earned=60;
              strcpy(Student_Phone, "123-1234 ");
            }
        if (Student_Number==2)
            { strcpy(Student_Name, "Anne");
              Credit_Earned=123;
              strcpy(Student_Phone, "123-2345 ");
            }
        if (Student_Number==3)

```

```

        { strcpy(Student_Name, "Greg      ");
          Credit_Earned=90;
          strcpy(Student_Phone, "123-7890 ");
        }

    }

// Next are methods of the STUDENT class ...

char *ReturnStudent_Name()
    {return Student_Name;};

char *ReturnStudent_Phone()
    {return Student_Phone;};

int ReturnCredit_Earned()
    {return Credit_Earned;};

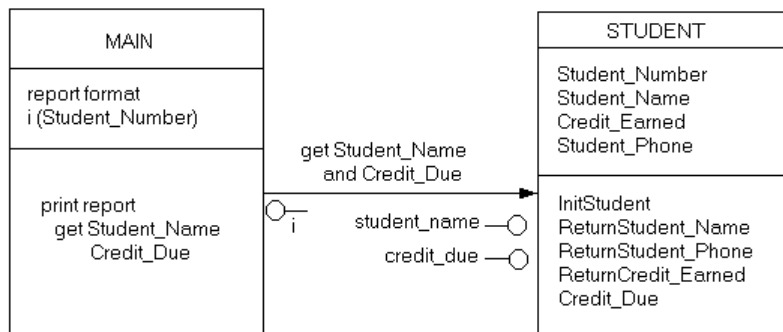
int Credit_Due() {
    int Due_Amount;
    if ((Credit_Earned)<120)
        { Due_Amount=120-Credit_Earned; }
    else
        { Due_Amount=0; };

    return(Due_Amount);
}
};

```

Q4-2. Draw an object-oriented analysis diagram for the above C++ program.

Answer:



Q4-3. Write the expected **print result** generated by the above C++ program.

Answer:

Student Name	Credits Needed for Graduation
John	60
Anne	0
Greg	30

=====

Q5

Q5-1. Read the following C++ program and complete it by filling the blanks.

```
#include<iostream>
#include<cstring>

#include_____

using namespace std;

void main()
{
    printf( "Used Car (Mileage)          List Price  \n");
    printf( "\n");

    for (int i=1; i<4; i++) {

        _____ Car;
        Car.InitCar(i);
        printf("%s  ",          Car.ReturnCar_Maker());
        printf("%u          ", Car.ReturnMileage());
        printf("$ %5.2f  \n", _____ .Car_Price());

    }

}

// **** Class CAR definition ****
```

```
// File is "car.h"

class CAR
{
    private:
        int      Stock_Number;
        char      Car_Maker[20];
        char      Car_Type_____;
        _____ Mileage;

    _____

    CAR()
    {
    };

// Operations

    void InitCar(int S_N) {
        Stock_Number=S_N;
        if (Stock_Number==1)
            { strcpy(Car_Maker, "Ford      ");
              strcpy(Car_Type, "Com");
              Mileage=80000;
            }
        if (Stock_Number==2)
            { strcpy(Car_Maker, "Toyota   ");
              strcpy(Car_Type, "Lux");
              Mileage=100000;
            }
        if (Stock_Number==3)
            { strcpy(Car_Maker, "Honda    ");
              strcpy(Car_Type, "Sml");
              Mileage=120000;
            }
    };

    char *ReturnCar_Maker()
        {return Car_Maker;};

    char ____ReturnCar_Type()
        {return Car_Type;};

    int ReturnMileage()
        {return Mileage;};

    double Car_Price() {
        double List_Price;
        if (strcmp(Car_Type, "Lux")==0)
            { List_Price=30000.0-Mileage*0.2; }
        else
```

```

        { List_Price=15000.0-Mileage*0.1; };

    return(List_Price);
};

```

Q5-2. Draw an object-oriented analysis diagram for the above C++ program.

Q5-3. Write the expected **print result** generated by the above C++ program.

Answers:

Q5-1 Answer:

```

#include<iostream>
#include<cstring>
#include"car.h"

using namespace std;

void main()
{
    printf( "Used Car (Mileage)           List Price  \n");
    printf( "\n");
    for (int i=1; i<4; i++) {

        CAR Car;
        Car.InitCar(i);
        printf("%s   ",      Car.ReturnCar_Maker());
        printf("%u           ", Car.ReturnMileage());
        printf("$ %5.2f  \n", Car.Car_Price());
    }
}

// **** Class CAR definition ****
// File is "car.h"

class CAR
{
    private:

        int      Stock_Number;
        char      Car_Maker[20];
        char      Car_Type[6];
        int      Mileage;

    public:
        CAR()
        {
            // void InitCar(int)    depending on S_N

```

```
};

// Operations

void InitCar(int S_N) {
    Stock_Number=S_N;
    if (Stock_Number==1)
        { strcpy(Car_Maker, "Ford    ");
          strcpy(Car_Type, "Com");
          Mileage=80000;
        }
    if (Stock_Number==2)
        { strcpy(Car_Maker, "Toyota  ");
          strcpy(Car_Type, "Lux");
          Mileage=100000;
        }
    if (Stock_Number==3)
        { strcpy(Car_Maker, "Honda   ");
          strcpy(Car_Type, "Sml");
          Mileage=120000;
        }
};

char *ReturnCar_Maker()
    {return Car_Maker;};

char *ReturnCar_Type()
    {return Car_Type;};

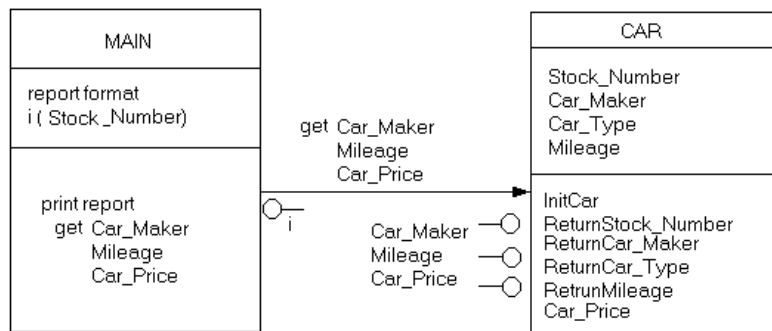
int ReturnMileage()
    {return Mileage;};

double Car_Price() {
    double List_Price;
    if (strcmp(Car_Type, "Lux")==0)
        { List_Price=30000.0-Mileage*0.2; }
    else
        { List_Price=15000.0-Mileage*0.1; };

    return(List_Price);
};
};
```

Q5-2. Draw an object-oriented analysis diagram for the above C++ program.

Answer:



Q5-3. Write the expected **print result** generated by the above C++ program.

Answer:

Used Car (Mileage)	List Price
Ford 80000	\$7000.0
Toyota 100000	\$10000.0
Honda 120000	\$3000.0

=====

Q6.

Q6-1. Read the following C++ program and complete it by filling the blanks.

```

#_____<iostream>
#include<cstring>

#include"emp.h"

using namespace std;

// The header file "emp.h" of the class EMPLOYEE used is included

void main()
_____

printf("Employee Name    Payment for Retire Fund\n");
printf("\n");

    for (int i=1; i<=3; i++) {

        EMPLOYEE Employee;
        _____.InitEmployee(i);
        printf ("%s    ", Employee.ReturnEmployee_Name());
        printf ("$$4.2f  \n", Employee.Retire_Fund());

    }
  
```



```

}

**** Class EMPLOYEE definition ****
// File is "emp.h"

class EMPLOYEE

// Class declaration
{
    _____:

// Attributes
    int            Employee_Number;
    char           Employee_Name[20];
    double         Employee_Salary;
    _____    Employee_Phone[15];

    public:
        EMPLOYEE ()
        {
// Constructor is actually implemented by the method
// void InitEmployee(int) depending on E_N
        };

// Operations

// The following procedure simulates the system to read a
// database/data file which records information of students

        void InitEmployee(int E_N) {
            Employee_Number=E_N;
            if (Employee_Number==1)
                { strcpy(Employee_Name, "Rick      ");
                  Employee_Salary=40000.0;
                  strcpy(Employee_Phone, "123-1234 ");
                }
            if (Employee_Number==2)
                { strcpy(Employee_Name, "Diane     ");
                  Employee_Salary=50000.0;
                  strcpy(Employee_Phone, "123-2345 ");
                }
            if (Employee_Number==3)
                { strcpy(Employee_Name, "Tim       ");
                  Employee_Salary=35000.0;
                  strcpy(Employee_Phone, "123-7890 ");
                }
        }

//Next are methods of the EMPLOYEE class ...

        _____ReturnEmployee_Name()

```

```

        {return Employee_Name;};

char *ReturnEmployee_Phone()
        {return Employee_Phone;};

double ReturnEmployee_Salary()
        {return Employee_Salary;};

        _____ Retire_Fund() {
            double Retire_Fund;
            if ((Employee_Salary)<45000.0)
                { Retire_Fund=Employee_Salary*0.1; }
            else
                { Retire_Fund=Employee_Salary*0.15; };
            _____(Retire_Fund);
        }
};

```

Q6-2. Draw an object-oriented analysis diagram for the above C++ program.

Q6-3. Write the expected **print result** generated by the above C++ program.

Answers:

Q6-1 Answer:

```

#include<iostream>
#include<cstring>

#include"emp.h"

using namespace std;

// The header file "emp.h" of the class EMPLOYEE used is included

void main()
{
    printf("Employee Name      Payment for Retire Fund\n");
    printf("\n");

    for (int i=1; i<=3; i++) {
        EMPLOYEE Employee;
        Employee.InitEmployee(i);
        printf ("%s      ",Employee.ReturnEmployee_Name());
        printf("$%4.2f  \n", Employee.Retire_Fund());
    }
}

```

```
// **** Class EMPLOYEE definition ****
// File is "emp.h"

class EMPLOYEE

// Class declaration
{
    private:

// Attributes
        int      Employee_Number;
        char      Employee_Name[20];
        double    Employee_Salary;
        char      Employee_Phone[15];

    public:
        EMPLOYEE ()
        {
// Constructor is actually implemented by the method
// void InitEmployee(int) depending on E_N
        };

// Operations

// The following procedure simulates the system to read a
// database/data file which records information of students

        void InitEmployee(int E_N) {
            Employee_Number=E_N;
            if (Employee_Number==1)
                { strcpy(Employee_Name, "Rick      ");
                  Employee_Salary=40000.0;
                  strcpy(Employee_Phone, "123-1234 ");
                }
            if (Employee_Number==2)
                { strcpy(Employee_Name, "Diane      ");
                  Employee_Salary=50000.0;
                  strcpy(Employee_Phone, "123-2345 ");
                }
            if (Employee_Number==3)
                { strcpy(Employee_Name, "Tim        ");
                  Employee_Salary=35000.0;
                  strcpy(Employee_Phone, "123-7890 ");
                }
        }

// Next are methods of the EMPLOYEE class ...

        char *ReturnEmployee_Name()
            {return Employee_Name;};
};
```

```

char *ReturnEmployee_Phone()
    {return Employee_Phone;};

double ReturnEmployee_Salary()
    {return Employee_Salary;};

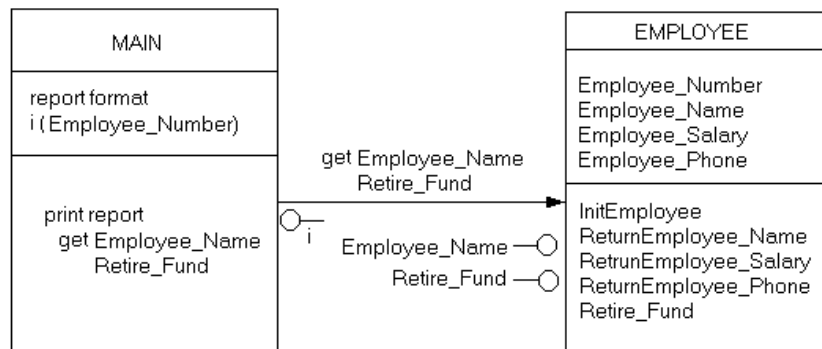
double Retire_Fund() {
    double Retire_Fund;
    if ((Employee_Salary)<45000.0)
        { Retire_Fund=Employee_Salary*0.1; }
    else
        { Retire_Fund=Employee_Salary*0.15; };

    return(Retire_Fund);
}
};

```

Q6-2. Draw an object-oriented analysis diagram for the above C++ program.

Answers:



Q6-3. Write the expected **print result** generated by the above C++ program.

Answers:

Employee Name	Payment for Retire Fund
Rick	\$4000.0
Diane	\$7500.0
Tim	\$3500.0