Question Bank

Q1) Write a C++ program to make a class Distance that has Centimeter and Meter, and then answer the following.

1- Set and print Function

2- Overload + to add two distance together.

3- Overload ++ to increase the Centimeter (100 centimeter is one meter)

```
class Distance {
   int Centimeter, Meter;
public:
Void set() {   Cin>> Centimeter>> Meter;   }
Void print()  {  Cout<< Centimeter<< Meter;   }
Distance operator+(Distance& d2)
   { Distance d3;
   d3. Centimeter = this-> Centimeter + d2. Centimeter;
   d3. Meter = this-> Meter + d2. Meter;
    if (d3.Centimeter >100)
         {   d3.Centimeter - = 100 ; d3.Meter ++ ;}
   return d3;
   }
void operator ++(){
++ Centimeter;
If (Centimeter >100)   {Centimeter =0 ; Meter++;}
}
void operator ++(int){   ++ Centimeter;
If (Centimeter >100)   {Centimeter =0 ; Meter++;}
}
};
int main()
{
Distance d1, d2, d3;
d1.set(); d2.set();
d3 = d1 + d2;
d3++;
d3.print();
   return 0;
}
```

Q2) Write a C++ program that consists of two classes, one for students (name, age, address, ID) and the second class for Marks (Mark1, Mark2, Mark3), then answer the following:

1- Inter the data of 25 students.

2- Print out the name and average of all students.

3- Print the name of students who has highest average

4- Print the name of students who fail in two Subjects.

```
class Mark{
   int Mark1, Mark2, Mark3;
    public:
Void set() {     Cin>> Mark1>> Mark2>> Mark3;   }
Void print()  {    Cout<< Mark1<< Mark2<< Mark3;   }
float Avg()  {return (Mark1 + Mark2 + Mark3)/3;}
};
class Student {
string name, address;
int  age, ID;
   Mark L;
public:
Void set() {  L.set();
                Cin>> name>>age>>address>> ID;   }
Void print()  {  L.print() ;
                  Cout<< name<<age<<address<<ID;   }
String getName() {return name;}
Float getAvg() { return L.Avg();}
};
int main()
{
Student X[25];
for(int i=0 ; i < 25 ; i++ )
X[i].set();
for(int i=0 ; i < 25 ; i++ )
cout<< X[i]. getName()<<X[i]. getAvg();
int A[25];
for(int i=0 ; i < 25 ; i++ )
A[i]=X[i].getAvg();
int Max=A[0], d=0;
for(int i=1 ; I < 25 ; i++ )
if (A[i] > Max) {Max =A[i]; d=0;}
cout<<X[d].getName()
 return 0;
}
```

Q) write a C++ program to make a class Date that has (Day, Month, Year) then answer the following

1- Constructor to accept Year by passing parameter and set day and month to 1
2- Write a friend function "Calculate" to calculate the number of days between two dates.
3- Overload (==) to compare two dates.
4- Overload (--) to decrease one month.

```
class Date{
   int Day, Month, Year;
    public:
Void set() {     Cin>> Day>> Month>> Year;   }
Void print()  {    Cout<< Day<< Month<< Year;   }
Date( int x){
Year= x;  day=month=1;}
Friend int calculate (date a1 , date a2);
Bool operator ==(date d1, date d2)
{return (d1. Day==d2. Day && d1. Month ==d2. Month && d1. Year ==d2. Year);

void operator -- (int){   Month--;
If (Month <=0)   {Month  =12 ; Year--;}
}
};
int calculate (date a1 , date a2)
{  int d1= a1.day – a2.day;
   int d2 = a1.Month – a2.Month ;
   int d3 = a1.Year – a2.Year;
return (d1 +d2*30 + (d3*12*30) );
}
int main()
{
Date X(198),Y(1975);
X.Set();
X.print()
Y.Set();
Y.print()
X--;
If (X==Y) cout<<:the two dates are equal";
   return 0;
}
```

Q) write a C++ program with class Rectangle_Paint (Width, Length), then answer the following

1- Default constructor and destructor.

2- Find the area of the rectangle.

3- Calculate the cost of painting the rectangle ($5 per m^2)

```
class Rectangle_Paint {
   int Width, Length;
    public:
Void set()
 {
   Cin>> Width >> Length;
  }
Void print()
{
   Cout <<Width,<<Length;
 }
Rectangle_Paint()
{
 Width = Length=0;
}
~Rectangle_Paint()
{
}
Int area() {
return Width * Length;
}
Int Calculate() {
 return area()*5;
}
};
int main()
{
Rectangle_Paint  X;
X.Set();
X.print()
Cout<<X.calculate();
   return 0;
}
```

Q) Write a C++ program to create class weight with gram and kilogram, then answer the following (1000 gram is 1 Kilogram)

1- Set and Print function.
2- Overload + to add to weights.
3- Overload ++ to increment gram.

```
    class weight {
       int gram , kilogram;
    public:
    Void set() {   Cin>> gram >> kilogram;   }
    Void print()  {  Cout<< gram << kilogram;   }
    weight operator+( weight & d2)
       { weight d3;
       d3. gram = this-> gram + d2. gram;
       d3. kilogram = this-> kilogram + d2. kilogram;
        if (d3.gram >1000)
             {   d3.gram - = 1000 ; d3. kilogram ++ ;}
       return d3;
       }
    void operator ++(){
    ++ gram;
    If (gram >1000)   {gram =0 ; Kilogram++;}
    }
    void operator ++(int){
    ++ gram;
    If (gram >1000)   {gram =0 ; Kilogram++;}
    }
    };
int main()
{
weight d1, d2, d3;
d1.set(); d2.set();
d3 = d1 + d2;
d3++;
d3.print();
   return 0;
}
```

Q) Write a C++ program that consists of two classes, one for Employee (name, age, address, gender) and the second class for Salary (basicSalary, Bonus, Tax), then answer the following

1- Inter the data of 70 Employees.

2- Print out the name and total salary of all employees.

(Total salary= basicSalary+ Bonus -Tax)

3- Print the name and age of employees who their total salaries are less than 1000$

4- Print the name of employees who will retired next year (retired year is 63)

```
class Salary {
   int basicSalary, Bonus, Tax;
    public:
Void set() {     Cin>> basicSalary>> Bonus>>Tax;   }
Void print()  {    Cout<< basicSalary<< Bonus<< Tax;   }
Int TotalSalary()  {return (basicSalary + Bonus -Tax);}
};
class Employee {
string name, address, gender;
int  age;
   Salary L;
public:
Void set() {  L.set();
                Cin>> name>>age>>address>> gender;   }
Void print()  {  L.print() ;
                  Cout<< name<<age<<address<<gender;   }
String getName() {return name;}
Float Tsalary () { return L. TotalSalary ();}
Int getage() {return age;}
};
int main()
{
Employee X[70];
For(int i=0 ; i < 70 ; i++ )
X[i].set();
For(int i=0 ; i < 70 ; i++ )
Cout<< X[i]. getName()<<X[i]. Tsalary ();
For(int i=0 ; i < 70 ; i++ )
If  (X[i]. Tsalary () < 1000)    {cout<<X[i].getname<<X[i].age()} else continue ;}
For(int i=0 ; i < 70 ; i++ )
If  (X[i]. getage () =62)    {cout<<X[i].getname();}
   return 0;
}
```

Q) write a C++ program to make a class Time that has (hour, Minute, Second) then answer the following

1- Constructor to accept minute and second by passing parameter and set hour to 12

2- Write a friend function "ZoneChange" to add two hours and 30 minutes.

3- Overload (--) to decrease one second.

4- Overload (==) to compare two times.

```
class Time {
   int hour, Minute, Second;
    public:
Void set() {     Cin>> hour>> Minute>> Second;   }
Void print()  {    Cout<< hour << Minute<<Second;   }
Time ( int x , int y){
Minute =  x;  second = Y ; Hour=12;}
Friend int ZoneChange (Time a1 , Time a2);
Bool operator ==( Time d1, Time d2)
{return(d1. Second ==d2. Second &&d1. Minute==d2. Minute&& d1. hour==d2. hour);
void operator -- (int){   Second--;
If (Second <0)   {Second  =0 ; Minute --;}
If (Minute <0 ) {Minute =0 ; Hour --;}
If (Hour <0 ) Hour =12;
}
};
Time ZoneChange (Time a1)
{  Time d;
d. minute =a1.minute + 30;
d.hour= a1.Hour + 2;
if(d.minute >60 ) {d.minute -= 60 ; d.hour++;}
if ( d.hour >12 ) { d.hour = d.hour  – 12 ; }
return  d;
}

int main()
{
Date X(5,6),Y(10,12);
X.Set();
X.print()
Y.Set();
Y.print()
X--;
If (X==Y) cout<<:the two Times are equal";
   return 0;
}
```

Q) write a C++ program with class BankAccount that has (Account_No, Balance) then answer the following

1- A constructor to reset the Balance of specific Account_NO and destructor
2- If the Account_No = 001 add $25 on the balance
3- A friend function to add $500 to all accounts, which is above that $2000.

```
class BankAccount {
    String Account_No,
   int Balance;
     public:
Void set()
 {
    Cin>> Account_No>> Balance;
  }
Void print()
{
    Cout << Account_No<< Balance;
 }
Void add(int X)
{ Balance +=X;}
BankAccount (int c)
{
 Account_NO = c;
Balance =0;
}
~ BankAccount ()
{
}
String getAccount() {return  Account_No;}
Friend void Add500(BankAccount &X);
};
Friend void Add500(BankAccount &X);
{  If (X. Balance>1000) X. Balance += 500; }
int main()
{
BankAccount  X;
X.Set();
If ( X. getAccount == "001") X.add(25);
X.print()
   return 0;
}
```

Q) What is OOP and way we use it?

- The essence of structured programming is to reduce a program into smaller parts and then code these elements more or less independently from each other. The structural model will help reusability and help dividing one problem to a number of smaller blocks each can be developed separately.

Q) Write a C++ program to inter 200 numbers then print all numbers who are above the average escaping zeros.

```
Void main() {
int a[200],d=0,Sum=0;
For(int i=0 ; i<200 ; i++ )
{cin>>a[1];
if(a[i] != 0) {d++; Sum+=a[i];}
}
Float avg=Sum/d;
For (int i=0 ; i<200 ; i++)
If (a[i]>avg) cout << a[i];
}
```

Q) Write a C++ program to create a "real_estate" class which consists of (**Type**, No_BedRoom, No_Bathroom, Rent_Price) the **Type** Only accept (House, Villa , Flat), then answer the following:

1- Create functions to set and print data,
2- Create a constructor to make all variables zero and one destructors.
3- Create a function "Show" to show all flats with rent price lower than 250$.
4- Create a function "Discount" to make a discount on Villas by 10%.

```
Class real_estate {
Private:
enum Type={ House, Villa , Flat};
int No_BedRoom, No_Bathroom, Rent_Price;
Public:
void set(){Cin>>Type>> No_BedRoom>>No_Bathroom>>Rent_Price;}
void print() {cout<<Type<< No_BedRoom<<No_Bathroom<< Rent_Price;}
real_estate () Type(House): No_BedRoom(0): No_Bathroom(0): Rent_Price(0){}
~real_estate(){}
Void show(){
if (Type==flat &&  Rent_Price<250) cout<< No_BedRoom<<No_Bathroom<< Rent_Price;}
void Discount (){ if(Type == villa) Rent_Price= Rent_Price- Rent_Price*10/100;}
}
int main()
{real_estate X[50];
for(int i=0 ; i<50 ; i++)
{X[i].set();
X[i].show();
X[i].Discount();
X[i].print();   }   }
```

Q) Write a C++ program to overload + to add two complex number like (5 + 2i and 7 + 6i) and – to subtract two complex number.

```
class Complex
{
      int num1, num2;
   public:
      void accept()
      {
            cout<<"\n Enter Two Complex Numbers : ";
            cin>>num1>>num2;
      }
Complex operator+(Complex obj)   //Overloading '+' operator
      {
            Complex c;
            c.num1=num1+obj.num1;
            c.num2=num2+obj.num2;
            return(c);
      }
Complex operator-(Complex obj)   //Overloading '+' operator
      {
            Complex c;
            c.num1=num1-obj.num1;
            c.num2=num2-obj.num2;
            return(c);
      }
      void display()
      {
            cout<<num1<<"+"<<num2<<"i"<<"\n";
      }
```

Q) Write a C++ program to create the following classes

Class Vehicle {string Name , int Cylinder , int **price**}{ set and print functions }

Class Show { int quantity , Tax is 5% if Cylinder =4 and 10% if Cylinder = 6 and Cost=**Price***quantity+Tax , set function which override vehicle set and print function which override vehicle print}

```
Class Show;
Class Vehicle{
protected:
string Name;
int Cylinder, Price;
public:
void set(){ cin>>Name>>Cylinder>>Price;}
Void print() {cout<< Name<<Cylinder<<Price;}
}
```

```
Class Show: Public Vehicle {
Private:
int quantity , Cost;
Public:
Void tax(){
If (Cylinder == 4)  Tax= Price *4/100; else if(Cylinder = =6) Tax=Price*6/100; else
tax=0;
Cost = price *quantity +Tax;}
Void set() {
Vehicle::set();
Cin>>quantity;}
Void print() {
Vehicle::Print();
Cout<<Quantity<<Cost
}
```

Q) Write three OOP properties.

- **Encapsulation**: the ability to deny a new type and a set of operations on that type, without revealing the representation of the type.
- **Inheritance**: the ability to create new types that inherit properties from existing types
- **Polymorphism**: the ability to use one name for two or more related but technically different purposes; one interface, multiple methods."

Q) Write a C++ to inter numbers and save it to array A[], stop interring when three zeros occurred and print the result.

```
Void main()
{
Int A[100] , x;
Int d=0 , i=0;
While(d <3)
{
Cin>>x;
If (x== 0) d++;
A[i++] = x;
}
For(int j=0 ; j<i ; j++)
Cout<<A[i];
}
```

Q) Write a C++ program to overload ++ to convert Fahrenheit to Celsius and vice versa

C = (F – 32) * 5 / 9

```
Class Conv {
Private:
int c , f , d=0;
public:
void set(){cin>>f ; cin >>c; }
Void operator ++() {
If (d==0) {c=(f-32)*5/9 ; d=1; cout<<"C="<<c;} else { f=c*9/5+32; d=0; cout<<"F= " <<f;}
}
Void print() {cout<<c<<f;}
```

Q) Write a C++ program that consists of the following class

1- Class "LIB" that hold the (Book_ID, Book_Name, Author, Publication, Quantity)
2- class "Borrow" consists of (Student_Name, Book_name, and Book_ID) and decrease the borrowed book from Quantity.

```
class LIB {
protected:
int Book_ID , Quantity ;
string Book_Name , Author , Publication ;
public:
void set() { cin>> Book_ID>>Book_Name>>Author>> Publication>> Quantity;}
void print() {cout<< Book_ID<<Book_Name<<Author<< Publication<< Quantity;}
}
Class Borrow : public LIB {
Private :
string  Student_Name:
pubic :
void set() { cin>> Student_name;}
void calc(string BN , string sn , int q ) {
if (BN == Book_Name) {Quantity -- ; Student _name=sn; cout<<Student_name
<<"took"<<Book_Name <<"and" <<Quantity <<"is left";}
}
}
```

Q) Write a C++ program to create the following classes Person { string name ,int Age, string Address , int fee } then:

1- write set and print function
2- create two constructor, one default that set everything to zero and one to set name as "Synonym" and Age as 18 if no parameter passed while declaration.
3- Create function Calculate so{if Address="Erbil" the fee=0 ; if age is under 30 the fee=0 , other cases fee=200$ }

```
class Person {

private:

string name ,  Address;

int Age  , fee;

void set(){ cin>> name >>  Address>> Age  >>fee;}

void print () { cout<< name <<  Address << Age << fee<<" $" ; }

Person () : name("0") ,  Address("0" ) , Age(0) , fee(0) { }

Person (string X = " Synonym " ,  int A = "18 ) : name (X), Age (A) { }

void Calculate () { if(Address == "Erbil" | | age<30 ) fee = 0; else fee = 200;}

}
```

Q) write the reasons that programmers prefer object programming.

- Object oriented programming is a new way to approach the task of programming.
- It supersedes the procedural or structured programming languages like Algol, Pascal or C
- The essence of structured programming is to reduce a program into smaller parts and then code these elements more or less independently from each other.

Q) Write a C++ program to create a "RealEstate" class which consists of (type, No_BedRoom, No_Bathroom, Rent_Price , Price) the type may be (House or Flat or any others), then answer the following:

1- Create functions to inter and print data.
2- Create a filter function to display houses which price is lower than 25000$.
3- Make a discount on Flats that has two bedroom by 5%.

```
Class real_estate {
Private:
string Type;
int No_BedRoom, No_Bathroom, Rent_Price,price ;
```

```
Public:
void set(){Cin>>Type>> No_BedRoom>>No_Bathroom>>Rent_Price>>price;}
void print() {cout<<Type<< No_BedRoom<<No_Bathroom<< Rent_Price<<price;}
Void filter(){
if (Type="hous" &&  price<25000)  print();}
void Discount (){ if(No_BedRoom==2 && Type="Flat") price= price- price*0.05 ;}
}
Int main()
{real_estate X[50];
for(int i=0 ; i<50 ; i++)
{X[i].set();
X[i].filter();
X[i].Discount();
X[i].print();
}
}
```

Q) Write a C++ program to inter length and width for a rectangle then find the area, use a class, find the area of 10 rectangles.

```
class Rectangle
{
    private:
        float width, length, area;
     public:
        void setData();
         void print();
         void calculate();
};
void Rectangle::setData()
{
   Cin>>width;
   Cin >>length;
}
void Rectangle::print()
{
Cout<<area;
}
void Rectangle:: calculate()
{ area = width * length;  }
int main()
{
  Rectangle rect [10];
For(int i=0 ; i<10 ; i++)
rect[i]. setData();
For(int i=0 ; i<10 ; i++)
```

```
{
        void calculate();
        void print();
}
   return 0;
}
```

Q) write a C++ program to convert Fahrenheit to Celsius using function.(C = (F– 32) *5/9)

```
double convert (int f)
{ return c=(f-32)*5/9 ; }
Void main()
{
int c , f;
cin>>f;
cout<<convert(f)
}
```

Q) Write a C++ program to make a class Distance that has Centimeter and Meter, and then answer the following.

1- Set and print Function

2- Overload + to add two distance together.

3- Overload ++ to increase the Centimeter (100 centimeter is one meter)

```
class Distance {
   int Centimeter, Meter;
public:
Void set() {   Cin>> Centimeter>> Meter;   }
Void print()  {  Cout<< Centimeter<< Meter;   }
Distance operator+(Distance& d2)
   { Distance d3;
   d3. Centimeter = this-> Centimeter + d2. Centimeter;
   d3. Meter = this-> Meter + d2. Meter;
    if (d3.Centimeter >100)
         {   d3.Centimeter - = 100 ; d3.Meter ++ ;}
   return d3;
   }
void operator ++(){
++ Centimeter;
If (Centimeter >100)   {Centimeter =0 ; Meter++;}
}
void operator ++(int){   ++ Centimeter;
If (Centimeter >100)   {Centimeter =0 ; Meter++;}
}
};
```

```
int main()
{
Distance d1, d2, d3;
d1.set(); d2.set();
d3 = d1 + d2;
d3++;
d3.print();
   return 0;
}
```

Q) Write a C++ program that consists of two classes, one for students (name, age, address, ID) and the second class for Marks (Mark1, Mark2, Mark3), then answer the following

1- Inter the data of 25 students.
2- Print out the name and average of all students.
3- Print the name of students who has highest average
4- Print the name of students who fail in two Subjects.

```
class Mark{
   int Mark1, Mark2, Mark3;
    public:
Void set() {     Cin>> Mark1>> Mark2>> Mark3;   }
Void print()  {    Cout<< Mark1<< Mark2<< Mark3;   }
float Avg()  {return (Mark1 + Mark2 + Mark3)/3;}
};
class Student {
string name, address;
int  age, ID;
   Mark L;
public:
Void set() {  L.set();
                Cin>> name>>age>>address>> ID;   }
Void print()  {  L.print() ;
                  Cout<< name<<age<<address<<ID;   }
String getName() {return name;}
Float getAvg() { return L.Avg();}
};
int main()
{
Student X[25];
For(int i=0 ; i < 25 ; i++ )
X[i].set();
For(int i=0 ; i < 25 ; i++ )
Cout<< X[i]. getName()<<X[i]. getAvg();
Int A[25];
For(int i=0 ; i < 25 ; i++ )
```

```
A[i]=X[i].getAvg();
Int Max=A[0], d=0;
For(int i=1 ; I < 25 ; i++ )
If (A[i] > Max) {Max =A[i]; d=0;}
Cout<<X[d].getName()

   return 0;
}
```

Q) write a C++ program to make a class Date that has (Day, Month, Year) then answer the following

1- Constructor to accept Year by passing parameter and set day and month to 1
2- Write a friend function "Calculate" to calculate the number of days between two dates.
3- Overload (==) to compare two dates.
4- Overload (--) to decrease one month.

```
class Date{
   int Day, Month, Year;
    public:
Void set() {     Cin>> Day>> Month>> Year;   }
Void print()  {    Cout<< Day<< Month<< Year;   }
Date( int x){
Year= x;  day=month=1;}
Friend int calculate (date a1 , date a2);
Bool operator ==(date d1, date d2)
{return (d1. Day==d2. Day && d1. Month ==d2. Month && d1. Year ==d2. Year);

void operator -- (int){   Month--;
If (Month <=0)   {Month  =12 ; Year--;}
}
};
int calculate (date a1 , date a2)
{  int d1= a1.day – a2.day;
   int d2 = a1.Month – a2.Month ;
   int d3 = a1.Year – a2.Year;
return (d1 +d2*30 + (d3*12*30) );
}

int main()
{
Date X(198),Y(1975);
X.Set();
X.print()
```

```
Y.Set();
Y.print()
X--;
If (X==Y) cout<<:the two dates are equal";
   return 0;
}
```

Q) write a C++ program with class Rectangle_Paint (Width, Length), then answer the following

1- Default constructor and destructor.

2- Find the area of the rectangle.

3- Calculate the cost of painting the rectangle ($5 per m^2)

```
class Rectangle_Paint {
   int Width, Length;
    public:
Void set()
 {     Cin>> Width >> Length;      }
Void print()
{       Cout <<Width,<<Length;      }
Rectangle_Paint()
{     Width = Length=0;     }
~Rectangle_Paint()
{}
Int area() {    return Width * Length;     }
Int Calculate() {    return area()*5;   }
};
int main()
{
Rectangle_Paint  X;
X.Set();
X.print()
Cout<<X.calculate();
   return 0;
}
```

Safeen H. Rasool