



Student Name: -

ID: -

Question 1: Chose the correct answer of the followings: 20 points

- 1- Regarding members of a class specified as Protected.
 - a) They can only be accessed by member functions of the class.
 - b) They can only be accessed by the member functions of class and friend functions.
 - c) They can only be accessed by member functions, friend functions and derived classes
 - d) All the previous answers are incorrect

- 2- If you have an object of class, which operator is used to access the object's members?
 - a) "."
 - b) "->"
 - c) "&"
 - d) none of the answers

- 3- Which operator used to get the address of an object ?
 - a) \$
 - b) =0
 - c) &
 - d) !

4. Which of the following is not a member of class?
 - A. Static method
 - B. Friend methods
 - C. const method
 - D. Virtual methods

5. How many instances of an abstract class can be created?
 - A. 0
 - B. 1
 - C. unlimited
 - D. non of the choices

6. Which of the following concepts provides facility of using a reference of an object inside another object?
 - A. Encapsulation
 - B. Abstraction
 - C. Aggregation
 - D. Inheritance

7. Which one of the following is the correct way to declare a pure virtual function?
 - A. virtual void Display(void){0};
 - B. virtual void Display = 0;
 - C. virtual void Display(void) = 0;
 - D. void Display(void) = 0;

8. Which of the following statement is correct?
 - A. Class is an instance of object.
 - B. Object is an instance of a class.
 - C. Class is an instance of data type.
 - D. Object is an instance of data type.

- 9 . an abstract class is a class that:
 - A. has at least one pure virtual method
 - B. all it methods are pure virtual methods.
 - C. have no data members.
 - D. have no method members.

- 10- Which statement is used to catch all types of exceptions?
 - a) catch()
 - b) catch(Test t)
 - c) catch(...)
 - d) none of the choices

Question 2: Given the following class that represent a vector in two dimension space **16 points**

```
class Vector
{
private:
    double X;
    double Y;
public:
    Vector(){X=Y=0.0;}
    Vector(double a,double b){X=a; Y=b;}
};
```

1. Define an << operator for output operation where the output should be in the form X_i+Y_j
2. Define * operatrion that represent dot product where the result is $C=AB$ where $C_x=A_x*B_x$ and $C_y=A_y*B_y$
3. Define ++ operator that increment X and Y by 1 unit (pre and post increment)
4. Define > operator where $A>B$ if $\sqrt{(A_x)^2 + (A_y)^2} > \sqrt{(B_x)^2 + (B_y)^2}$

```
class Vector
{
    friend ostream& operator<<(ostream& out,Vector& T);
    friend Vector operator*(Vector& T1,Vector& T2);
private:
    double X;
    double Y;
public:
    Vector(){X=Y=0.0;}
    Vector(double a,double b){X=a; Y=b;}
    Vector operator++()
    {
        X=X+1;
        Y=Y+1;
        return *this;
    }
    Vector operator++(int u)
    {
        Vector T=*this;
        X=X+1;
        Y=Y+1;
        return T;
    }
    bool operator>(Vector T)
    {
        if(sqrt(pow(X,2.0)+pow(Y,2.0))>sqrt(pow(T.X,2.0)+pow(T.Y,2.0)))
            return true;
        return false;
    }
};
ostream& operator<<(ostream& out,Vector& T)
{
    out<<T.X<<"i+"<<T.Y<<"j";
    return out;
}
Vector operator*(Vector& T1,Vector& T2)
{
    Vector T;
    T.X=T1.X*T2.X;
    T.Y=T1.Y*T2.Y;
    return T;
}
```

Question 3: A students data are stored in a sequential access file where the first record represent the number of students. And the remaining records represent the information of students as the following ID FName LName GPA separated by white spaces as shown in the example below

16 points

| | |
|-----------------------|-------------------|
| 2 | Number of records |
| 1001 Ahmad Hani 90.5 | Record 1 |
| 1002 Sami Sameer 80.3 | Record 2 |

Write a C++ program that perform the followings:

- 1- define a structure to store one student information.
- 2- Open the file above and read the first record.
- 3- Create a dynamically allocated array of student structure.
- 4- Read the data from the file to Array created in previous section.

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

struct STUDENT{
    int ID;
    string FNAME;
    string LNAME;
    float GPA;
};

int main()
{
    int Size;
    STUDENT* std;
    ifstream F("file.txt",ios::in);
    F>>Size;
    std=new STUDENT[Size];
    for(int i=0;i<Size;i++)
        F>>std[i].ID>>std[i].FNAME>>std[i].LNAME>>std[i].GPA;

    F.close();

    return 0;
}
```

Question 4: Given the following program:

16 points

```
struct Date
{
    int Day;
    int Month;
    int Year;
};

void Set_Date(Date& Dt,int D,int M,int Y)
{
    Dt.Day=D;
    Dt.Month=M;
    Dt.Year=Y;
}

int main()
{
    Date Dt;
    int D,M,Y;
    cout<<"Day: ";    cin>>D;
    cout<<"Month: ";  cin>>M;
    cout<<"Year: ";   cin>>Y;

    Set_Date(Dt,D,M,Y);

    return 0;
}
```

The value of the year should not be negative.

The value of the month should be between 1 and 12.

The value of the Day should be greater than 0 and the upper bound depends on the month and year.

- 1- Define an exception handling class that is thrown in function Set_Date with the appropriate message if one of the above conditions is violated.
- 2- The function Set_Date should rethrow the exception to caller function.
- 3- Define try catch block in main that catches the exception if occur.

```
class BAD_DATE_INFO:public exception
{
private:
    string MESSAGE;
public:
    BAD_DATE_INFO():exception("Bad Date Info"){MESSAGE="Bad Date Info";}
    BAD_DATE_INFO(string n):exception(n.c_str()){MESSAGE=n;}
    string what(){return MESSAGE;}
};
```

```

struct Date
{
    int Day;
    int Month;
    int Year;
};

void Set_Date(Date& Dt,int D,int M,int Y) throw(BAD_DATE_INFO)
{
    if(Y<0)
        throw BAD_DATE_INFO("Year must be positive integer");
    Dt.Year=Y;
    if(M>12||M<1)
        throw BAD_DATE_INFO("Month must be between 1 and 12");
    Dt.Month=M;
    if(D<=0)
        throw BAD_DATE_INFO("Day must be positive integer");
    if((M==4||M==6||M==9||M==11)&&D>30)
        throw BAD_DATE_INFO("Day must be less than 31");
    if((M==1||M==3||M==5||M==7||M==8||M==10||M==12)&&D>31)
        throw BAD_DATE_INFO("Day must be less than 32");
    if(M==2&&(Y%4!=0)&&D>28)
        throw BAD_DATE_INFO("Day must be less than 29");
    if(M==2&&(Y%4==0)&&D>29)
        throw BAD_DATE_INFO("Day must be less than 30");
    Dt.Day=D;
}

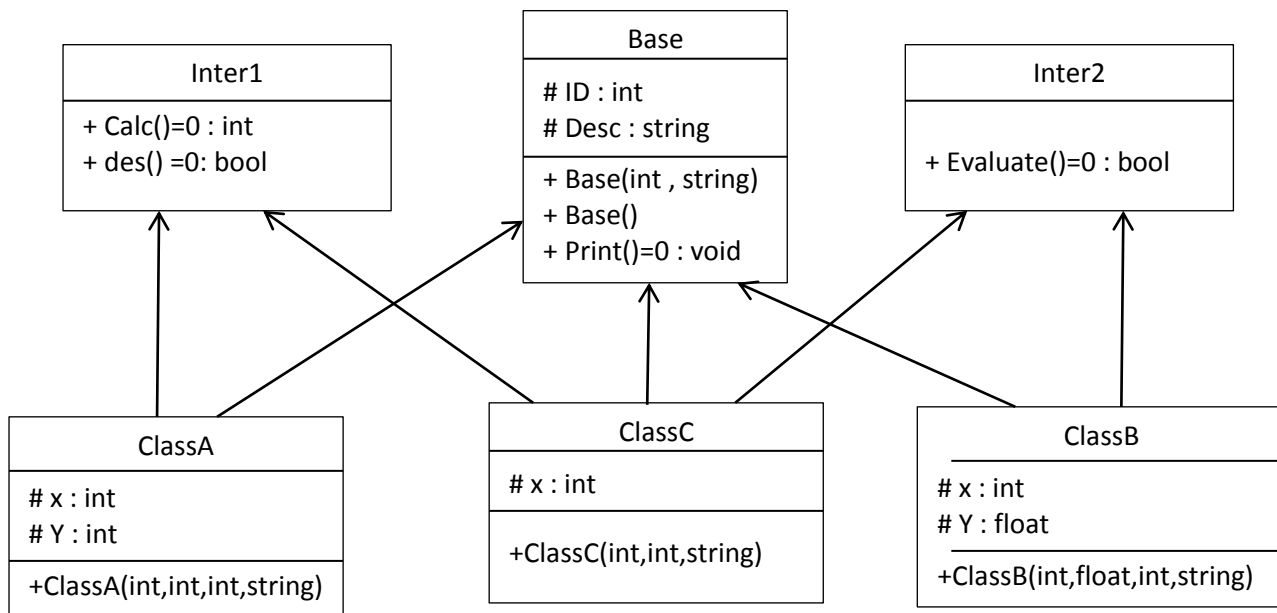
int main()
{
    Date Dt;
    int D,M,Y;
    cout<<"Day: ";    cin>>D;
    cout<<"Month: ";  cin>>M;
    cout<<"Year: ";   cin>>Y;
    try{
        Set_Date(Dt,D,M,Y);
    }
    catch(BAD_DATE_INFO e)
    {
        cout<<"ERROR: "<<e.what()<<endl;
    }

    return 0;
}

```

Question 5: Given the following UML diagram, Construct the class hierarchy shown bellow.

20 points



```

class Inter1
{
public:
    virtual int Calc()=0;
    virtual bool dec()=0;
};
class Inter2
{
public:
    virtual bool Evaluate()=0;
};
class Base
{
protected:
    int ID;
    string Desc;
public:
    Base(int x,string s){ID=x; Desc=s;}
    Base(){ID=0; Desc="";}
    virtual void Print()=0;
};
class ClassA:public Base,public Inter1
{
protected:
    int x,y;
public:
    ClassA(int a,int b,int z,string s):Base(z,s){x=a; y=b;}
    void Print()
    {
        cout<<"x="<<x<<"\ty="<<y<<"\tID="<<ID<<"\tDesc: "<<Desc;
    }
    int Calc()
    
```

```

    {
        return x*y;
    }
bool dec()
{
    if(x==y)
        return true;
    return false;
}
};
class ClassB:public Base,public Inter2
{
protected:
    int x;
    float y;
public:
    ClassB(int a,float b,int c,string s):Base(c,s){x=a; y=b;}
    void Print()
    {
        cout<<"x="<<x<<"\ty="<<y<<"\tID="<<ID<<"\tDesc: "<<Desc;
    }
    bool Evaluate()
    {
        if(x==ID)
            return true;
        return false;
    }
};
class ClassC:public Base,public Inter1,public Inter2
{
protected:
    int x;
public:
    ClassC(int a,int b,string s):Base(b,s){x=a;}
    void Print()
    {
        cout<<"x="<<x<<"\tID="<<ID<<"\tDesc: "<<Desc;
    }
    bool Evaluate()
    {
        if(x==ID)
            return true;
        return false;
    }
    int Calc()
    {
        return x;
    }
    bool dec()
    {
        if(x==ID)
            return true;
        return false;
    }
};

```

Question 6: Define a class that have the following properties: **12 points**

- 1- allow only 3 object to be allocated at the same time.
- 2- If there is 3 object in memory it will deny creating new object.
- 3- If an object in memory was deleted then an new object can be created.

```
class Test
{
private:
    Test();
    static int COUNT;
public:
    static Test* Init()
    {
        Test* T;
        if(COUNT<3)
        {
            T=new Test;
            return T;
        }
        else
            return null;
    }
    ~Test() {COUNT--;}
};
int Test::COUNT=0;
```