Name and Title : Student Evaluation Portal

In C++ a virtual method is a method of the base class that is expected to be overridden in a derived class.

A pure virtual function is a method of a class that has no definition making it **'abstract'.** One or more pure virtual functions, when added to a class, effectively make the said class 'Abstract Base Class' ensuring instantiation i.e creation of objects is not possible for the class.

Requirements: Create a student evaluation portal that manages marks & grades calculation for Engineering & Architecture branch students of a college

A class named Student has the following members:

Data Member Name	Data Type	Description
StudentName	string	Stores the name of the Student.
StudentId	string	Stores the ID of the Student.
TotalMarks	int	Stores the total marks of the student. This member must be accessible in the immediate derived class of Student.

Complete all the below member functions of the Employee class as per the given instructions:

Method Name	Arguments	Return Type	Description
Default & parameterized constructor	none	none	Constructors to initialize Student objects (with 0 arguments and with parameters) respectively.
Destructor	none	none	Method is used to destroy the object at the end of scope.
GetStudentName	none	string	Returns the name of the Student
GetStudentIId	none	string	Returns the ID of the Student
GetTotalMarks	none	int	Returns total marks of the student.
CalculateMarks	none	none	A pure virtual function to be overridden in derived classes for calculating total marks of students.

CalculateGrade	none	char	Calculates grade to be allotted to the student based on total marks as per conditions specified below

Grade Calculation Formula

- If TotalMarks is equal to -1, the grade returned should be '-' (hyphen character indicating invalid exam attempt).
- If TotalMarks is over 100, the grade returned should be '#" (Pound sign or hash character indicating invalid total marks calculation).
- If TotalMarks is between 85 and 100 (both inclusive), the grade returned should be 'O'.
- If TotalMarks is between 61 and 84 (both inclusive), the grade returned should be 'A'.
- If TotalMarks is between 40 and 60 (both inclusive), the grade returned should be 'P'.
- If Student scores below 40, the grade should be'F'.

Derived Class 'EngineeringStudent'

A class named ***EngineeringStudent*** is an extension (inherits) of the Employee class, has the following member variables:

Data Member Name	Data Type	Description
PracticalMarks	int	Stores the marks obtained by the student in practical exams.
TheoryMarks	int	Stores the marks obtained by the student in theory exams.

Complete all the below member functions of the Manager class as per the given instructions:

Method Name	Arguments	Return Type	Description
Default & parameterized constructor	none	none	Default constructor is used to create an **EngineeringStudent** object with no initial value. Parameterized constructor can take values for all data members defined above to initialize the Manager object. (Note: Base class data members also need to be initialized while creating Manager objects).

Destructor	none	none	The method is used to destroy the object at the end of the scope.
GetPracticalMarks	none	int	Return the marks obtained by the student in practical exams.
GetTheoryMarks	none	int	Returns the marks obtained by the student in theory exams.
GetTotalMarks	none	int	Returns the total marks obtained by the student by adding marks from theory and practical exams together.
CalculateMarks	none	void	Sets the value of *TotalMarks* data member(inherited from Student base class) by adding marks from theory and practical exam together.

Derived Class 'ArchitectureStudent'

A class named ***ArchitectureStudent*** is an extension (inherits) of the Employee class, has the following member variables:

Data Member Name	Data Type	Description
DesignMarks	int	Stores the marks obtained by the student in architectural designs exams.
ThesisMarks	int	Stores the marks obtained by the student in thesis project..

Complete all the below member functions of the Manager class as per the given instructions:

Method Name	Arguments	Return Type	Description
Default & parameterized constructor	none	none	Default constructor is used to create an **EngineeringStudent** object with no initial value. Parameterized constructor can take values for all data members defined above to initialize the Manager object. (Note: Base class data members also need to be initialized while creating Manager objects).
Destructor	none	none	The method is used to destroy the object at the end of the scope.
GetDesignMarks	none	int	Return the marks obtained by the student in design exams.
GetThesisMarks	none	int	Returns the marks obtained by the student in thesis exams.
GetTotalMarks	none	int	Returns the total marks obtained by the student by adding marks from design and thesis project marks together.
CalculateMarks	none	void	Sets the value of *TotalMarks* data member(inherited from Student base class) by adding marks from thesis and design exams together.

Please note that you will find class declarations in “bitmap.h” and to complete class definition you will add code in file “bitmap.cpp”