A. The College Member Class

CollegeMember
protected int mID_Number protected String mFirstName, mLastName, mGender protected int mDateOfBirth protected String mTelephone, mEmail public static final int DEFAULT_ID = 0
public CollegeMember() public CollegeMember(CollegeMember thisMember) public void modifyMe(CollegeMember thisMember) public String printMe() public void inputData(int x, String inCategory) public String toString() public int getID() protected void finalize() public boolean equals(Object thisObject) public int compareTo(Object thisObject)

A1. The CollegeMember() Constructor

```
START
    mID_Number, mDateOfBirth := DEFAULT_ID;
    mFirstName, mLastName, mGender, mTelephone, mEmail := " ";
STOP
```

A2. The CollegeMember(CollegeMember thisMember) Overloaded Constructor

```
START
    mID_Number := thisMember.mID_Number;
    mLastName := thisMember.mLastName;
    mFirstName := thisMember.mFirstName;
    mGender := thisMember.mGender;
    mDateOfBirth := thisMember.mDateOfBirth;
    mEmail := thisMember.mEmail;
    mTelephone := thisMember.mTelephone;
STOP
```

A3. The void modifyMe(CollegeMember thisMember) Method

```
START
    mID_Number := thisMember.mID_Number;
    mLastName := thisMember.mLastName;
    mFirstName := thisMember.mFirstName;
    mGender := thisMember.mGender;
    mDateOfBirth := thisMember.mDateOfBirth;
    mEmail := thisMember.mEmail;
    mTelephone := thisMember.mTelephone;
STOP
```

A4. The int getID() Method

START

Return mID_Number;

STOP

A5. The void inputData(int x, String inCategory) Method

START

Let inputID, inputTele, inputFName, inputLName, inputDoB be strings;
Prompt for and accept inputID for inCategory x;
While (NOT validateID(inputID)) do the following:
DisplayMessage("ID Number must be numeric");
Prompt for and accept inputID for inCategory x;
End-While;
Prompt for and accept inputFName for inCategory x;
While (inputName' s first character is not a letter) do the following:
DisplayMessage("Name must begin with a letter");
Prompt for and accept inputFName for inCategory x;
End-While;
Prompt for and accept inputLName for inCategory x;
While (inputName' s first character is not a letter) do the following:
DisplayMessage("Name must begin with a letter");
Prompt for and accept inputLName for inCategory x;
End-While;
Prompt for and accept mEmail for inCategory x
Prompt for and accept mGender for inCategory x;
Prompt for and accept inputTele for inCategory x;
While (NOT validateTele(inputTele)) do the following:
DisplayMessage("Telephone number is not in the required format");
Prompt for and accept inputTele for inCategory x;
End-While;
Prompt for and accept inputDoB for inCategory x;
While (NOT validateDoB(inputDoB)) do the following:
DisplayMessage("Invalid Date of Birth");
Prompt for and accept inputDoB for inCategory x;
End-While;
mID_Number := inputID;
mFirstName := inputFName;
mLastName := inputLName;
mTelephone := inputTele;
mDateOfBirth := inputDoB;

STOP

A6. The boolean validateTele(String thisTele) Method

START

Let isValid be Boolean, initialized to true;
Let x be an integer;

```
        For (x:= 1 to 12 with increments of 1) do the following:
                Case x is
                4, 8: If (thisTele.CharacterAt(x-1) <> '-') isValid := false; EndIf;
                Otherwise: If (thisTele.CharacterAt(x-1) is not a digit) isValid := false; EndIf;
                End-Case;
        End-For
        Return isValid;
STOP
        A7. The boolean validateID(String thisID) Method
START
        Let isValid be Boolean, initialized to true;
        Let x be an integer;
        For (x:= 1 to thisID.length() with increments of 1) do the following:
                If (thisID.CharacterAt(x-1) is not a digit) isValid := false; EndIf;
        End-For;
        Return isValid;
STOP
        A8. The boolean validateDoB(String thisDate) Method
START
        Let isValid be Boolean, initialized to True;
        Let x, Year, Month, Day be integers;
        Let mCheck be an array of 13 integers;
        Let LeapYear be a Boolean flag;
        Let CurrentYear be the current year as retrieved from the system;
        Set Year to Substring(thisDate, 0,4);
        Set Month to Substring(thisDate,4,2);
        Set Day to Substring(thisDate, 6,2);
        LeapYear:= False;
        If (Year mod 400) is 0) OR ((Year Mod 4) is 0 AND (Year Mod 100) <> 0)) LeapYear :=
True; End-If;
        mCheck[0] := 0; mCheck[1] := 31; mCheck[2] := 28; If (LeapYear ) mCheck[2] := 29;
End-If
        mCheck[3], mCheck[5], mCheck[7], mCheck[8]; mCheck[10], mCheck[12] := 31;
        mcheck[4], mCheck[6], mCheck[9], mCheck[11] := 30;
        If (Year > CurrentYear) isValid := False; End-If;
        Else If (Month < 1 OR Month > 12) isValid := False; End-If;
                Else If (Day > mCheck[Month]) isValid := False; End-If;
        Return isValid;
STOP
        A9. The String printMe( ) Method
START
        Let printString be a string;
```

printString := "ID Number: " + mID_Number +" Name: " + mFirstName + " "+ mLastName + + "Gender: " + mGender + ", " + " Date of Birth: " + mDateOfBirth + + "mTelephone: " + mTelephone + + "E-Mail: " + mEmail;
Return printString;
STOP

A10. The void finalize() Method
START
Destroy the current object and call the garbage collection routine;
STOP

A11. The String toString() Method
START
Return "CollegeMember: " + this.printMe();
STOP

A12. The boolean equals(Object thisObject) Method
START
Let instanceMatch be Boolean, initialized to false;
If (thisObject is an instance of CollegeMember)
    Let thisMember be a CollegeMember object, instantiated by thisObject;
    If (mID_Number = thisMember.mID_Number) instanceMatch := true; End-If;
End-If;
Return instanceMatch;
STOP

B. The Generic Class

Generic
protected String alumAcadDept = Null, alumMajor= Null protected String alumCurrentEmployer= Null, alumJobTitle= Null protected String stdAcadDept= Null, stdAcadMajor= Null protected String empJobDept= Null, empJobSpecialization= Null, empJobTitle= Null Protected int setDataType Protected String dataTypeString = “”
public Generic(CollegeMember thisMember) public void modifyMe(Generic thisItem) public void inputData(int x) public String printMe() public String toString() protected void finalize() public boolean equals(Object thisObject) public void setDataType(int dataType)

B1. The Generic(CollegeMember thisMember) Constructor
START
super (thisMember);

```
        //based on which data Item is being chosen
        Case setDataType is
                1: empJobDept, empJobSpecialization, empJobTitle:= " ";
                2:stdAcadDept, stdAcadMajor:= " ";
                3:alumAcadDept , alumMajor, alumCurrentEmployer, alumJobTitle:= " ";
        End-Case;
STOP
```

B2. The void modifyMe(Generic thisItem) Method

```
START
        Let x be an integer;
        super.modifyMe(thisItem);
        Case setDataType is
                1:empJobDept := thisItem.empJobDept; empJobSpecialization:=
        thisEmp.empJobSpecialization; empJobTitle:= thisEmp.empJobTitle;
                2:stdAcadDept := thisItem.stdAcadDept; stdAcadMajor:= thisItem.stdAcadMajor;
                3:alumAcadDept  := thisAlum.alumAcadDept ; alumMajor:= thisAlum.alumMajor;
alumCurrentEmployer := thisAlum.alumCurrentEmployer; alumJobTitle:= thisAlum.alumJobTitle;
        End-Case;
STOP
```

B3. The public void inputData(int x) Method

```
START
        super.inputData(x, dataTypeString);
        Case setDataType is:
                1:Prompt for and accept empJobDept for employee x; Prompt for and accept
empJobSpecialization for employee x; Prompt for and accept empJobTitle for employee x;
                2:Prompt for and accept stdAcadDept for student x; Prompt for and accept
stdAcadMajor for student x;
                3:Prompt for and accept alumAcadDept for alumnus x; Prompt for and accept
alumMajor for alumnus x; Prompt for and accept alumCurrentEmployer  for alumnus x; Prompt
for and accept alumJobTitle for alumnus x;
STOP
```

B4. The public String printMe() Method

```
START
        Let printString be a string;
        Let x be an integer;
        Case setDataType is:
                1:printString := super.printMe() + + "Department: " + Department + +
"Specialization: " + Specialization + + "Job Title: " + jobTitle;
                2:printString := super.printMe() + + "Department: " + acadDept + + "Major: " +
acadMajor;
```

3:printString := super.printMe() + + "Department: " + acadDept + + "Major: " + acadMajor; + + “Current Employer “ + currentEmployer + + “Job Title: “ + jobTitle;
Return printString;
STOP

B5. The public String toString() Method
START
Return dataTypeString + this.printMe();
STOP

B6. The protected void finalize() Method
START
Destroy the current object and call the garbage collection routine;
STOP

B7. The public boolean equals(Object thisObject) Method
START
Let instanceMatch be Boolean, initialized to false;
If (thisObject is an instance of Generic)
Let thisMember be a Generic object, instantiated by thisObject;
If (mID_Number = thisMember.mID_Number) instanceMatch := true; End-If;
End-If;
Return instanceMatch;
STOP

B8. The void setDataType(int dataType) Method
START
setDataType = dataType
Case dataType is:
1:dataTypeString = “Employee”
2:dataTypeString = “Student”
3:dataTypeString = “Alumnus”
STOP

C. The CollegeHashController Class

CollegeHashController
public static Hashtable <Integer, Generic> studentsList; public static Hashtable <Integer, Generic> employeesList; public static Hashtable <Integer, Generic> alumnusList; public static ArrayList keyValues; public static final String HEADING = "CollegeMember Hash Table"; public static final int DEFAULT_NUMBER = 0;

```
Public static void main(String[] args)
Public static void inputStudents()
Public static void inputEmployees()
Public static void inputAlumnus()
Public static void queryStudents()
Public static void queryEmployees()
Public static void queryAlumnus()
Public static void listStudents()
Public static void listEmployees()
Public static void listAlumnus()
Public static void removeStudents()
Public static void removeEmployees()
Public static void removeAlumnus()
Public static void checkStudentSize()
Public static void checkEmployeeSize()
Public static void checkAlumniSize()
Public static void initializeLists()
Public static void emptyStudents()
Public static void emptyEmployees()
Public static void emptyAlumnus()
```

C0. The main(String[] args) Method

```
START
        Declare String promptString to your menu options
        Declare boolean exitTime to false
        Declare int nextUserAction, userOption

        Call initializeLists()
        While(exitTime=false)
                Present promptString as menu and accept userOption as input
                switch(userOption)
                        case 0:{exitTime = true; break;}
                        case 1:{inputStudents();break;}
                        case 2:{inputEmployees();break;}
                        case 3:{inputAlumnus();break;}
                        case 4:{queryStudents();break;}
                        case 5:{queryEmployees();break;}
                        case 6:{queryAlumnus();break;}
                        case 7:{listStudents();break;}
                        case 8:{listEmployees();break;}
                        case 9:{listAlumnus();break;}
                        case 10:{removeStudents();break;}
                        case 11:{removeEmployees();break;}
                        case 12:{removeAlumnus();break;}
                        case 13:{checkStudentSize();break;}
                        case 14:{checkEmployeeSize();break;}
```

```
                    case 15:{checkAlumniSize();break;}
                    case 16:{emptyStudents();break;}
                    case 17:{emptyEmployees();break;}
                    case 18:{emptyAlumnus();break;}
            End switch
        End while
STOP

C1. The void inputStudents() Method
START
        Declare int numberOfStudents, x, studentsListSize
        Declare currentStudent as an instance of Generic
        Set studentsListSize equal to studentsList.size()
        Accept input for numberOfStudents
        ensureCapacity for keyValues for studentsListSize plus numberOfStudents
        for(x = 1 to numberOfStudents incrementing by 1) do the following
                Set currentStudent equal to a new Generic
                setDataType of currentStudent to 2
                InputData at position x for currentStudent
                Put currentStudent in studentsList
                Add currentStudent.getID to keyValues at position x-1
        End for
STOP

C2. The void inputEmployees() Method
START
        Declare int numberOfEmployees, x, employeesListSize
        Declare currentEmployee as an instance of Generic
        Set employeesListSize equal to employeesList.size()
        Accept input for numberOfEmployees
        ensureCapacity for keyValues for employeesListSize plus numberOfEmployees
        for(x = 1 to numberOfEmployees incrementing by 1) do the following
                Set currentEmployee equal to a new Generic
                setDataType of currentEmployee to 1
                InputData at position x for currentEmployee
                Put currentEmployee in employeeList
                Add currentEmployee.getID to keyValues at position x-1
        End for
STOP

C3. The void inputAlumnus() Method
START
        Declare int numberOfAlumnus, x, alumnusListSize
        Declare currentAlumnus as an instance of Generic
```

```
        Set alumnusListSize equal to alumnusList.size()
        Accept input for numberOfAlumnus
        ensureCapacity for keyValues for alumnusListSize plus numberOfAlumnus
        for(x = 1 to numberOfAlumnus incrementing by 1) do the following
                Set currentAlumnus equal to a new Generic
                setDataType of currentAlumnus to 3
                InputData at position x for currentAlumnus
                Put currentAlumnus in employeeList
                Add currentAlumnus.getID to keyValues at position x-1
        End for
STOP
```

C4. The void queryStudents() Method

```
START
        Declare String outString
        Declare int nextUserAction, searchKey
        Let soughtStudent be a Generic instance
        Declare String qHeading to “Student Hash Table Query”
        Declare boolean exitNow to false

        while(exitNow equals false) do the following
                Accept input for searchKey
                Let soughtStudent be a new Generic instance
                Set soughtStudent dataType to 2
                if (studentsList contains searchKey)
                        Call modifyMe for soughtStudent at searchKey
                        printMe for soughtStudent to outString
                        Print outString
                End if
                Else
                        outString equals “Student specified is not in the list”
                        Print outString
                End else
                Accept input for if user would like continue
        End while
STOP
```

C5. The void queryEmployees() Method

```
START
        Declare String outString
        Declare int nextUserAction, searchKey
        Let soughtEmployee be a Generic instance
        Declare String qHeading to “Employee Hash Table Query”
        Declare boolean exitNow to false
```

```
        while(exitNow equals false) do the following
                Accept input for searchKey
                Let soughtEmployee be a new Generic instance
                Set soughtEmployee dataType to 1
                if (employeesList contains searchKey)
                        Call modifyMe for soughtEmployee at searchKey
                        printMe for soughtEmployee to outString
                        Print outString
                End if
                Else
                        outString equals “Employee specified is not in the list”
                        Print outString
                End else
                Accept input for if user would like continue
        End while
STOP
```

C6. The void queryAlumnus() Method

```
START
        Declare String outString
        Declare int nextUserAction, searchKey
        Let soughtAlumnus be a Generic instance
        Declare String qHeading to “Alumnus Hash Table Query”
        Declare boolean exitNow to false

        while(exitNow equals false) do the following
                Accept input for searchKey
                Let soughtAlumnus be a new Generic instance
                Set soughtAlumnus dataType to 3
                if (alumnusList contains searchKey)
                        Call modifyMe for soughtAlumnus at searchKey
                        printMe for soughtAlumnus to outString
                        Print outString
                End if
                Else
                        outString equals “Alumnus specified is not in the list”
                        Print outString
                End else
                Accept input for if user would like continue
        End while
STOP
```

C7. The void listStudents() Method

START
	Let x and studentListSize be integers
	Set studentListSize equal to studentsList.size()
	Declare String outString equal to “The Hash table contains the following:”
	for(x = 1 to studentsListSize incrementing by 1) do the following
		Add studentsList.get(X) to outString
	End for
	Print outString
STOP

C8. The void listEmployees() Method
START
	Let x and employeeListSize be integers
	Set employeeListSize equal to employeeList.size()
	Declare String outString equal to “The Hash table contains the following:”
	for(x = 1 to employeeListSize incrementing by 1) do the following
		Add employeeList.get(X) to outString
	End for
	Print outString
STOP

C9. The void listAlumnus() Method
START
	Let x and alumnusListSize be integers
	Set alumnusListSize equal to alumnusList.size()
	Declare String outString equal to “The Hash table contains the following:”
	for(x = 1 to alumnusListSize incrementing by 1) do the following
		Add alumnusList.get(X) to outString
	End for
	Print outString
STOP

C10. The void removeStudents() Method
START
	Let removalPrompt be a string
	Declare String Heading to “Removal of Students from the Hash Table”
	Let x, removalKey and nextUserAction be integers
	Declare boolean exitNow to false

	While(exitNow == false) do the following
		Accept input for removalKey
		while(removalKey is not in studentsList) do the following
			Show error message
			Accept input for removalKey

```
            End while
            Declare removalPrompt to “Student” + removalKey + " is about to be
removed from the hash table.\n" + "Click Yes to remove the items." + "Click No or Cancel
to exit."
            Accept input for nextUserAction
            if (nextUserAction == Yes) do the following
                  Remove item at removalKey from studentsList
                  Remove item at removalKey from keyValues array
            End if
            Ask the user if they want to continue
      End while
STOP

C11. The void removeEmployees() Method
START
      Let removalPrompt be a string
      Declare String Heading to “Removal of Employees from the Hash Table”
      Let x, removalKey and nextUserAction be integers
      Declare boolean exitNow to false

      While(exitNow == false) do the following
            Accept input for removalKey
            while(removalKey is not in employeesList) do the following
                  Show error message
                  Accept input for removalKey
            End while
            Declare removalPrompt to “Employee” + removalKey + " is about to be
removed from the hash table.\n" + "Click Yes to remove the items." + "Click No or Cancel
to exit."
            Accept input for nextUserAction
            if (nextUserAction == Yes) do the following
                  Remove item at removalKey from employeesList
                  Remove item at removalKey from keyValues array
            End if
            Ask the user if they want to continue
      End while
STOP

C12. The void removeAlumnus() Method
START
      Let removalPrompt be a string
      Declare String Heading to “Removal of Alumnus from the Hash Table”
      Let x, removalKey and nextUserAction be integers
      Declare boolean exitNow to false
```

```
        While(exitNow == false) do the following
                Accept input for removalKey
                while(removalKey is not in alumnusList) do the following
                        Show error message
                        Accept input for removalKey
                End while
                Declare removalPrompt to “Alumni” + removalKey + " is about to be
removed from the hash table.\n" + "Click Yes to remove the items." + "Click No or Cancel
to exit."
                Accept input for nextUserAction
                if (nextUserAction == Yes) do the following
                        Remove item at removalKey from alumnusList
                        Remove item at removalKey from keyValues array
                End if
                Ask the user if they want to continue
        End while
STOP

C13. The void checkStudentSize() Method
START
        Show a message dialog containing the following:
        “There are “ + studentsList.size() + " students in the hash table",HEADING, +
JOptionPane.INFORMATION_MESSAGE
STOP

C14. The void checkEmployeeSize() Method
START
        Show a message dialog containing the following:
        “There are “ + employeesList.size() + " employees in the hash table",HEADING,
+ JOptionPane.INFORMATION_MESSAGE
STOP

C15. The void checkAlumniSize() Method
START
        Show a message dialog containing the following:
        “There are “ + alumnusList.size() + " alumnus in the hash table",HEADING, +
JOptionPane.INFORMATION_MESSAGE
STOP

C16. The void initializeLists() Method
START
        Declare studentsList to a new Hashtable
        Declare employeesList to a new Hashtable
```

```
        Declare alumnusList to a new Hashtable
        Declare keyValues to a new ArrayList
STOP
```

C17. The void emptyStudents() Method

```
START
        Let x, nextUserAction be integers
        Declare String removalPrompt to “You are about to empty the hash table” + "Click
Yes to Empty. Click No or Cancel to exit"
        Accept input for nextUserAction
        if(nextUserAction == yes) do the following
                Clear studentsList
                Clear keyValues
        End if
STOP
```

C18. The void emptyEmployees() Method

```
START
        Let x, nextUserAction be integers
        Declare String removalPrompt to “You are about to empty the hash table” + "Click
Yes to Empty. Click No or Cancel to exit"
        Accept input for nextUserAction
        if(nextUserAction == yes) do the following
                Clear employeesList
                Clear keyValues
        End if
STOP
```

C19. The void emptyAlumnus() Method

```
START
        Let x, nextUserAction be integers
        Declare String removalPrompt to “You are about to empty the hash table” + "Click
Yes to Empty. Click No or Cancel to exit"
        Accept input for nextUserAction
        if(nextUserAction == yes) do the following
                Clear alumnussList
                Clear keyValues
        End if
STOP
```