Name: **Blake Allard**
Class: **MW 8am**

BEGIN

```
INITIALIZE  totalCandidates   = 0
INITIALIZE  candidateCount   = 0

FUNCTION  sex = ValidateSex()

        DO
                INPUT         sex

                IF      (!(cin.get(sex)))
                        cin.clear();
                END IF

                PROCESSING      invalidSex = sex != 'x' && sex != 'X' &&
                                             sex != 'f' && sex != 'F' &&
                                             sex != 'm' && sex != 'M';
                IF      invalidSex
                        OUTPUT      "ERROR: INVALID sex - Please Input M/F"
                END IF
                WHILE       invalidSex

                RETURN      sex

WHILE       (sex != 'x' && sex != 'X')


        FUNCTION        height = ValidateIntInput       (INVALID_MIN_HEIGHT,
                                                         INVALID_MAX_HEIGHT,
                DO                                       Height: , "INVALID! ",
                        INPUT       intInput             Please enter a height between );

                        IF      (!(cin >> intInput))
                                cin.clear();

                        END IF

                        IF      intInput < minValue || intInput > maxValue
                                OUTPUT      errPrompt;
                        ELSE
                                validInput = false;
                WHILE       validInput    invalidprmpt;

                RETURN      intInput;
```

```
FUNCTION        weight = ValidateIntInput       (INVALID_MIN_WEIGHT,
                                                INVALID_MAX_WEIGHT,
    DO                                          Weight: , "INVALID! ",
        INPUT       intInput                    Please enter a Weight between );

        IF      (!(cin >> intInput))
                cin.clear();
                OUTPUT
        END IF

        IF      intInput < minValue || intInput > maxValue
                OUTPUT      errPrompt;
        ELSE
                validInput = invalidprmpt;
    WHILE       validInput

    RETURN      intInput;


PROCESSING      ++totalCandidates


FUNCTION        acceptableHeight = CheckMeasurement  (sex , height)

    RETURN      ((sex == 'm' || sex == 'M')
                && intInput >= MIN_RANGE_M && intInput <= MAX_RANGE_M) ||
                ((sex == 'f' || sex == 'F')
                && intInput >= MIN_RANGE_F && intInput <= MAX_RANGE_F);


FUNCTION        acceptableWeight = CheckMeasurement  (sex , weight)

    RETURN      ((sex == 'm' || sex == 'M')
                && intInput >= MIN_RANGE_M && intInput <= MAX_RANGE_M) ||
                ((sex == 'f' || sex == 'F')
                && intInput >= MIN_RANGE_F && intInput <= MAX_RANGE_F);


FUNCTION        EvaluateCandidate(acceptableHeight, acceptableWeight, acceptedCount);

    IF          acceptableHeight && acceptableWeight
        OUTPUT      "This candidate has been ACCEPTED!";
    END IF

    ELSE IF     acceptableWeight
        OUTPUT      "This candidate has been rejected based on the HEIGHT requirements";
    END IF
```

```
        ELSE IF     acceptableHeight
            OUTPUT      "This candidate has been rejected based on the WEIGHT requirements";
        END IF

        ELSE
            OUTPUT      "This candidate has been rejected based on the HEIGHT and WEIGHT requirements";
        END IF


    FUNCTION        sex = ValidateSex()

        DO
            INPUT       sex

            IF      (!(cin.get(sex)))
                    cin.clear();
            END IF

            PROCESSING      invalidSex = sex != 'x' && sex != 'X' &&
                                         sex != 'f' && sex != 'F' &&
                                         sex != 'm' && sex != 'M';
            IF      invalidSex
                    OUTPUT      "ERROR: INVALID sex - Please Input M/F"
            END IF
            WHILE       invalidSex

            RETURN      sex

END WHILE

IF      totalCandidates > 0

        PROCESSING      percent = (double(acceptedCount) / totalCandidates) * 100;

        OUTPUT      acceptedCount
        OUTPUT      percent
END IF

END
```

VARIABLE LIST

INPUT	OUTPUT (IN LOOP)	PROCESSING (IN LOOP)
sex	EvalutateCandidate	ValidateIntInput
height	OUTPUT (OUT OF LOOP)	totalCandidates
weight	acceptedCount	acceptableHeight
	percent	acceptableWeight
		EvaluateCandidate
		PROCESSING (OUT OF LOOP)
		percent

FUNCTIONS LIST

```
char ValidateSex          (const int PROMPT_SIZE);

int ValidateIntInput      (int minValue,
                           int maxValue,
                     const int PROMPT_SIZE,
                           string prompt1
                           string errprmpt
                           string invalidPrmpt);

bool CheckMeasurement     (char charInput,
                           int intInput,
                           const int MIN_RANGE_A
                           const int MAX_RANGE_A
                           const int MIN_RANGE_B
                           const int MAX_RANGE_B);

void  EvaluateCandidate   (bool acceptableHeight,
                           bool acceptableWeight,
                           int acceptedCount);
```

```
 1 ****************************************************
 2 *  PROGRAMMED BY :  Blake Allard
 3 *  CLASS         :  CS1A
 4 *  SECTION       :  M/W 8am
 5 *  ASSIGNMENT #6 :  Military Academy
 6 ****************************************************
 7
 8 Please enter the candidate's information (enter 'X' to exit).
 9 Sex:     m
10 Height:  80
11 Weight:  130
12
13 This candidate has been ACCEPTED!
14
15
16 Please enter the candidate's information (enter 'X' to exit).
17 Sex:     m
18 Height:  65
19 Weight:  250
20
21 This candidate has been ACCEPTED!
22
23
24 Please enter the candidate's information (enter 'X' to exit).
25 Sex:     f
26 Height:  62
27 Weight:  109
28
29 This candidate has been rejected based on the WEIGHT requirements.
30
31
32 Please enter the candidate's information (enter 'X' to exit).
33 Sex:     x
34
35 2 candidate(s) accepted!
36 That's 67%!
37
38
39
```

```
 1 ***************************************************
 2 *  PROGRAMMED BY :  Blake Allard
 3 *  CLASS         :  CS1A
 4 *  SECTION       :  M/W 8am
 5 *  ASSIGNMENT #6 :  Military Academy
 6 ***************************************************
 7
 8 Please enter the candidate's information (enter 'X' to exit).
 9 Sex:     M
10 Height:  65
11 Weight:  250
12
13 This candidate has been ACCEPTED!
14
15
16 Please enter the candidate's information (enter 'X' to exit).
17 Sex:     m
18 Height:  64
19 Weight:  130
20
21 This candidate has been rejected based on the HEIGHT requirements.
22
23
24 Please enter the candidate's information (enter 'X' to exit).
25 Sex:     M
26 Height:  80
27 Weight:  251
28
29 This candidate has been rejected based on the WEIGHT requirements.
30
31
32 Please enter the candidate's information (enter 'X' to exit).
33 Sex:     male
34 Height:  81
35 Weight:  129
36
37 This candidate has been rejected based on the HEIGHT and WEIGHT requirements.
38
39
40 Please enter the candidate's information (enter 'X' to exit).
41 Sex:     F
42 Height:  75
43 Weight:  110
44
45 This candidate has been ACCEPTED!
46
47
48 Please enter the candidate's information (enter 'X' to exit).
49 Sex:     f
50 Height:  76
51 Weight:  185
```

```
52
53 This candidate has been rejected based on the HEIGHT requirements.
54
55
56 Please enter the candidate's information (enter 'X' to exit).
57 Sex:     female
58 Height:  62
59 Weight:  109
60
61 This candidate has been rejected based on the WEIGHT requirements.
62
63
64 Please enter the candidate's information (enter 'X' to exit).
65 Sex:     F
66 Height:  61
67 Weight:  186
68
69 This candidate has been rejected based on the HEIGHT and WEIGHT requirements.
70
71
72 Please enter the candidate's information (enter 'X' to exit).
73 Sex:     F
74 Height:  63
75 Weight:  110
76
77 This candidate has been ACCEPTED!
78
79
80 Please enter the candidate's information (enter 'X' to exit).
81 Sex:     x
82
83 3 candidate(s) accepted!
84 That's 33%!
85
86
87
```

```
 1 ***************************************************
 2 *  PROGRAMMED BY :  Blake Allard
 3 *  CLASS         :  CS1A
 4 *  SECTION       :  M/W 8am
 5 *  ASSIGNMENT #6 :  Military Academy
 6 ***************************************************
 7
 8 Please enter the candidate's information (enter 'X' to exit).
 9 Sex:     x
10
```

```
/*****************************************************************************
 * AUTHOR     : Blake Allard
 * STUDENT ID : 358888
 * ASN #6     : Military Academy
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/18/24
 *****************************************************************************/

#ifndef AS6_H_
#define AS6_H_

#include <iostream>    /* cout, cin           */
#include <iomanip>     /* setw                */
#include <string>      /* string              */
using namespace std;

    /*************************************************************************
     * OutputClassHeader
     *   This function receives an assignment name, type and number then outputs
     *   the appropriate class header.
     *   asType is defaulted to Lab as there are more labs than Assignments.
     *-----------------------------------------------------------------------
     *   ==> returns nothing - This will output the class heading.
     *************************************************************************/

    void OutputClassHeader();                    // OUT - output class header

    /*************************************************************************
     * ValidateSex
     *   This function receives a letter input representing the user's sex,
     *   validates if the character input is 'm', 'M', 'f', or 'F'.  If not
     *   the user will be prompted with an error message and asked to reenter a
     *   valid sex.
     *-----------------------------------------------------------------------
     *   ==> returns sex - sex will be used for determining the candidate's
     *       acceptance status dependent upon either sex's specific number range
     *************************************************************************/

    char ValidateSex(const int PROMPT_SIZE);       // OUT - column width size

    /*************************************************************************
     * ValidateIntInput
     *   This function receives a min & max value, a constant prompt size for
     *   output formatting, a string prompt, an error message prompt if input is
     *   determined as invalid, and an invalid prompt if the invalid number
     *   entered is not within the mix & max value range.
     *-----------------------------------------------------------------------
     *   ==> returns intInput - this will be used for determining a candidate's
     *       eligibility dependent upon the sex's matching min/max number range
     *************************************************************************/
```

```
52
53      int  ValidateIntInput (int    minValue     ,   //     / OUT - minimum value
54                             int    maxValue     ,   //     / OUT - maximum value
55                       const int    PROMPT_SIZE  ,   // IN  & OUT - prompt size
56                             string prompt1      ,   // IN  & OUT - first prompt
57                             string errprmpt     ,   // IN  & OUT - error prompt
58                             string invalidPrmpt);   // IN  & OUT - invalid #
59
60      /*************************************************************************
61       * CheckMeasurement
62       *   This function receives a min & max value, a constant prompt size for
63       *   output formatting, a string prompt, an error message prompt if input is
64       *   determined as invalid, and an invalid prompt if the invalid number
65       *   entered is not within the mix & max value range.
66       *------------------------------------------------------------------------
67       *   ==> returns intInput - this will be used for determining a candidate's
68       *       eligibility dependent upon the sex's matching min/max number range
69       *************************************************************************/
70
71      bool CheckMeasurement (char charInput  ,       // IN   /      - sex input
72                             int  intInput   ,       // IN   /      - measurement
73                       const int  MIN_RANGE_A ,      //      / OUT  - min range
74                       const int  MAX_RANGE_A ,      //      / OUT  - max range
75                       const int  MIN_RANGE_B ,      //      / OUT  - min range
76                       const int  MAX_RANGE_B);      //      / OUT  - max range
77
78      /*************************************************************************
79       * EvaluateCandidate
80       *   This function receives an acceptableHeight boolean variable & an
81       *   acceptableWeight boolean variable, and returns whether or not both,
82       *   one, or none of the variables are true or false based on each boolean
83       *   variable's compound comparison statement, then the function outputs
84       *   a string response dependent on which variables are true and/or false.
85       *------------------------------------------------------------------------
86       *   ==> returns nothing - this will be used for determining a candidate's
87       *       eligibility response dependent upon their character & integer
88       *       inputs that are paired with their matching ranges.
89       *************************************************************************/
90
91      void EvaluateCandidate(bool acceptableHeight ,  // IN   & OUT  - eval height
92                             bool acceptableWeight);  // IN   & OUT  - eval weight
93
94
95  #endif
96
```

```cpp
/*****************************************************************************
 * AUTHOR     : Blake Allard
 * STUDENT ID : 358888
 * ASN #6     : Military Academy
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/18/24
 *****************************************************************************/

#include "as6.h"       /* header file         */

/*****************************************************************************
 * PROGRAM DESCRIPTION
 * ---------------------------------------------------------------------------
 * This program will determine the acceptance of a male or female candidate
 * based on the candidate's height & weight
 * ---------------------------------------------------------------------------
 * INPUT: the user will input the candidate's sex, height & weight.
 *
 * OUTPUT (in loop): based on the candidates eligibility the program will
 *                   output:
 *
 *     - "This candidate has been ACCEPTED!"
 *     - "This candidate has been rejected based on the HEIGHT requirement."
 *     - "This candidate has been rejected based on the WEIGHT requirement."
 *     - "This candidate has been rejected based on the HEIGHT and WEIGHT "
 *       "requirements."
 *
 *
 * Example Input/Output:
 *
 * Please enter the candidate’s information (enter ‘X’ to exit).
 * Sex: m
 * Height: 80
 * Weight: 130
 *
 * This candidate has been ACCEPTED!
 *****************************************************************************/

int main()
{

   /**************************************************************************
    * CONSTANTS
    * ------------------------------------------------------------------------
    * FORMATTING - USED FOR PROMPT COLUMN SIZE
    * ------------------------------------------------------------------------
    * PROMPT_SIZE : used for setting the prompt column size
    * ------------------------------------------------------------------------
    * PROCESSING - USED FOR MALE & FEMALE MIN/MAX VALID HEIGHT AND WEIGHT
    * ------------------------------------------------------------------------
```

```
 52     * M_MIN_RANGE_1 - MINIMUM VALID HEIGHT FOR MEN
 53     * M_MAX_RANGE_1 - MAXIMUM VALID HEIGHT FOR MEN
 54     * F_MIN_RANGE_2 - MINIMUM VALID HEIGHT FOR WOMEN
 55     * F_MAX_RANGE_2 - MAXIMUM VALID HEIGHT FOR WOMEN
 56     * M_MIN_RANGE_3 - MINIMUM VALID WEIGHT FOR MEN
 57     * M_MAX_RANGE_3 - MAXIMUM VALID WEIGHT FOR MEN
 58     * F_MIN_RANGE_4 - MINIMUM VALID WEIGHT FOR WOMEN
 59     * F_MAX_RANGE_4 - MAXIMUM VALID WEIGHT FOR WOMEN
 60     *************************************************************************/
 61
 62     //CONSTANTS - column width size
 63     const int PROMPT_SIZE = 9;
 64
 65     //CONSTANTS - min/max ranges that determine acceptance per category
 66     const int M_MIN_RANGE_HEIGHT  = 65;
 67     const int M_MAX_RANGE_HEIGHT  = 80;
 68     const int F_MIN_RANGE_HEIGHT  = 62;
 69     const int F_MAX_RANGE_HEIGHT  = 75;
 70     const int M_MIN_RANGE_WEIGHT  = 130;
 71     const int M_MAX_RANGE_WEIGHT  = 250;
 72     const int F_MIN_RANGE_WEIGHT  = 110;
 73     const int F_MAX_RANGE_WEIGHT  = 185;
 74
 75    /*************************************************************************
 76     * CONSTANTS
 77     * -----------------------------------------------------------------------
 78     * PROCESSING - USED FOR BOUNDARY VALUES TO ERROR CHECK HEIGHT & WEIGHT INPUT
 79     * -----------------------------------------------------------------------
 80     * VALID_MIN_HEIGHT - VALID MINIMUM HEIGHT
 81     * VALID_MAX_HEIGHT - VALID MAXIMUM HEIGHT
 82     * VALID_MIN_WEIGHT - VALID MINIMUM WEIGHT
 83     * VALID_MAX_WEIGHT - VALID MAXIMUM WEIGHT
 84     *************************************************************************/
 85
 86     //CONSTANTS - boundary values for invalid input
 87     const int VALID_MIN_HEIGHT = 24;
 88     const int VALID_MAX_HEIGHT = 110;
 89     const int VALID_MIN_WEIGHT = 50;
 90     const int VALID_MAX_WEIGHT = 1400;
 91
 92     /*************************************************************************
 93      * VARIABLES -
 94      *************************************************************************/
 95
 96     char   sex;                 // IN   & CALC - sex of candidate
 97     int    height;              // IN   & CALC - candidates height
 98     int    weight;              // IN   & CALC - candidate's weight
 99     int    totalCandidates;     // CALC &      - total # candidates evaluated
100     int    acceptedCount;       // CALC & OUT  - # of accepted candidates
101     double percent;             // CALC & OUT  - % of accepted candidates
102     bool   acceptableHeight;    // CALC &      - min/max height requirements
```

```
103     bool   acceptableWeight;      // CALC &      - min/max weight requirements
104
105     /*************************************************************************
106      * INITIALIZATIONS -
107      ************************************************************************/
108
109     totalCandidates = 0;          // CALC &      - total candidates evaluated
110     acceptedCount   = 0;          // CALC & OUT  - number of accepted candidates
111
112     /*************************************************************************
113      * OUTPUT - class heading
114      ************************************************************************/
115
116     OutputClassHeader();
117
118     /*************************************************************************
119      * INPUT - prompt the user for the candidate's sex, height & weight
120      * ----------------------------------------------------------------------
121      * EXAMPLE:
122      *         Please enter the candidate’s information (enter ‘X’ to exit).
123      *         Sex:    m
124      *         Height: 80
125      *         Weight: 130
126      ************************************************************************/
127
128     cout << left;
129
130     sex = ValidateSex(PROMPT_SIZE);
131
132     /*************************************************************************
133      * PROCESSING - validates sex and validates height & weight acceptability
134      ************************************************************************/
135
136
137     while (sex != 'x' && sex != 'X')
138     {
139
140         height = ValidateIntInput(VALID_MIN_HEIGHT,
141                                   VALID_MAX_HEIGHT, PROMPT_SIZE,
142                                   "Height: " , "\nINVALID! MUST BE A NUMBER!\n",
143                                   "\nINVALID! Please enter a height between ");
144
145         weight = ValidateIntInput(VALID_MIN_WEIGHT,
146                                   VALID_MAX_WEIGHT, PROMPT_SIZE,
147                                   "Weight: " , "\nINVALID! MUST BE A NUMBER!\n",
148                                   "\nINVALID! Please enter a weight between ");
149
150         cout << endl;
151
152
153         acceptableHeight = CheckMeasurement(sex , height ,
```

```
154                                             M_MIN_RANGE_HEIGHT ,
155                                             M_MAX_RANGE_HEIGHT ,
156                                             F_MIN_RANGE_HEIGHT ,
157                                             F_MAX_RANGE_HEIGHT);
158
159         acceptableWeight = CheckMeasurement(sex , weight ,
160                                             M_MIN_RANGE_WEIGHT ,
161                                             M_MAX_RANGE_WEIGHT ,
162                                             F_MIN_RANGE_WEIGHT ,
163                                             F_MAX_RANGE_WEIGHT);
164
165
166         //INCREMENT - increment total candidate count
167         ++totalCandidates;
168
169
170         EvaluateCandidate(acceptableHeight, acceptableWeight);
171
172         //INCREMENT - increments accepted count if height & weight acceptable
173         if (acceptableHeight && acceptableWeight)
174         {
175             acceptedCount++;
176         }
177
178
179         sex = ValidateSex(PROMPT_SIZE);
180
181
182     } //...end while (sex != 'x')
183
184
185     /*************************************************************************
186     * OUTPUT - output the # of candidates accepted & the percentage of accepted
187     *------------------------------------------------------------------------
188     * EXAMPLE:
189     *          3 candidate(s) accepted!
190     *          That's 33%!
191     *************************************************************************/
192
193
194     if (totalCandidates > 0)
195     {
196
197         percent = (double(acceptedCount) / totalCandidates) * 100;
198
199         cout << endl;
200
201         cout << setprecision(0) << fixed;
202
203
204         cout << acceptedCount << " candidate(s) accepted!\n";
```

```
205         cout << "That's "     << percent << "%!\n\n\n";
206
207
208         cout << setprecision(6);
209         cout.unsetf(ios::fixed);
210
211         cout << right;
212     }
213
214     return 0;
215
216 } //... end int main()
217
```

```cpp
/*****************************************************************************
 * AUTHOR     : Blake Allard
 * STUDENT ID : 358888
 * ASN #6     : Military Academy
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/18/24
 *****************************************************************************/

#include "as6.h"       /* header file         */

/*****************************************************************************
 * CONSTANTS
 * ---------------------------------------------------------------------------
 * OUTPUT - USED FOR CLASS HEADING
 * ---------------------------------------------------------------------------
 * PROGRAMMER      : Programmer's Name
 * CLASS           : Student's Course
 * SECTION         : Class Day / Times
 * EXR/ASN/LAB_NUM : Exercise/Assignment/Lab #
 * EX/ASN/LAB_NAME : Title of the Exercise/Assignment/Lab
 *****************************************************************************/

//OUTPUT - used for class heading
const char PROGRAMMER[] = "Blake Allard";
const char CLASS[]      = "CS1A";
const char SECTION[]    = "M/W 8am";
const char ASN_NUM[]    = "6";
const char ASN_NAME[]   = "Military Academy";


void OutputClassHeader()
{
    cout <<  left;
    cout <<  "*********************************************************\n";
    cout <<  "*  PROGRAMMED BY :  "<< PROGRAMMER                      << "\n";
    cout <<  "*  "      << setw(14)<< "CLASS"   << ":  " << CLASS    << "\n";
    cout <<  "*  "      << setw(14)<< "SECTION" << ":  " << SECTION  << "\n";
    cout <<  "*  ASSIGNMENT #" << setw(2) << ASN_NUM   << ":  " << ASN_NAME <<
        "\n";
    cout <<  "*********************************************************\n\n";
    cout <<  right;
}

```

```
/*****************************************************************************
 * AUTHOR     : Blake Allard
 * STUDENT ID : 358888
 * ASN #6     : Military Academy
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/18/24
 *****************************************************************************/

#include "as6.h"       /* header file         */

/*****************************************************************************
 * ValidateSex
 *   This function receives a letter input representing the user's sex,
 *   validates if the character input is 'm', 'M', 'f', or 'F'.  If not
 *   the user will be prompted with an error message and asked to reenter a
 *   valid sex.
 *---------------------------------------------------------------------------
 *   ==> returns sex - sex will be used for determining the candidate's
 *       acceptance status dependent upon either sex's specific number range
 *****************************************************************************/

    char ValidateSex(const int PROMPT_SIZE) // IN       - column width size

{

    char sex;                               // IN & OUT - obtains and validates
                                            //            character input
    bool invalidSex;                        // IN       - initializes invalidSex
                                            //            to error check input


    do

    {
        cout << "Please enter the candidate's information (enter 'X' to exit)."
            "\n";
        cout << setw(PROMPT_SIZE) << "Sex: ";
        if (!(cin.get(sex)))
        {
            cin.clear();
        }

        invalidSex = sex != 'x' && sex != 'X' &&
                     sex != 'f' && sex != 'F' &&
                     sex != 'm' && sex != 'M';

        if (invalidSex)
        {
            cout << "\nERROR: INVALID sex - Please Input M/F\n";
        }
```

```
52
53          cin.ignore(10000, '\n');
54
55      }while (invalidSex);
56
57      return sex;
58
59  }
60
```

```
 1 /*****************************************************************************
 2  * AUTHOR     : Blake Allard
 3  * STUDENT ID : 358888
 4  * ASN #6     : Military Academy
 5  * CLASS      : CS1A
 6  * SECTION    : MW 8am
 7  * DUE DATE   : 11/18/24
 8  *****************************************************************************/
 9
10 #include "as6.h"       /* header file         */
11
12 /*****************************************************************************
13  * ValidateIntInput
14  *   This function receives a min & max value, a constant prompt size for
15  *   output formatting, a string prompt, an error message prompt if input is
16  *   determined as invalid, and an invalid prompt if the invalid number
17  *   entered is not within the mix & max value range.
18  *---------------------------------------------------------------------------
19  *   ==> returns intInput - this will be used for determining a candidate's
20  *       eligibility dependent upon the sex's matching min/max number range
21  *****************************************************************************/
22
23    int ValidateIntInput(int    minValue ,    //       OUT - minimum value
24                         int    maxValue ,    //       OUT - maximum value
25                   const int    PROMPT_SIZE , // IN  & OUT - prompt size
26                         string prompt1 ,     // IN  & OUT - first prompt
27                         string errPrompt ,   // IN  & OUT - error prompt
28                         string invalidprmpt) // IN  & OUT - invalid #
29
30 {
31          int  intInput;                      //       OUT - number input
32          bool validInput;                    //       IN  - check validInput
33
34    validInput = true;
35
36    do
37    {
38        cout << setw(PROMPT_SIZE) << prompt1;
39        if (!(cin >> intInput))
40        {
41            cin.clear();
42
43            cout << errPrompt;
44        }
45
46        else if (intInput < minValue || intInput > maxValue)
47        {
48            cout << invalidprmpt;
49            cout << minValue << " and " << maxValue << endl;
50        }
51        else
```

```
52          {
53              validInput = false;
54          }
55
56          cin.ignore(10000, '\n');
57
58      }while(validInput);
59
60      return intInput;
61  }
62
```

```cpp
/*****************************************************************************
 * AUTHOR     : Blake Allard
 * STUDENT ID : 358888
 * ASN #6     : Military Academy
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/18/24
 *****************************************************************************/

#include "as6.h"       /* header file         */

/*****************************************************************************
 * CheckMeasurement
 *   This function receives a min & max value, a constant prompt size for
 *   output formatting, a string prompt, an error message prompt if input is
 *   determined as invalid, and an invalid prompt if the invalid number
 *   entered is not within the mix & max value range.
 *---------------------------------------------------------------------------
 *   ==> returns intInput - this will be used for determining a candidate's
 *       eligibility dependent upon the sex's matching min/max number range
 *****************************************************************************/

    bool CheckMeasurement (char sex        ,   // IN        - sex input
                           int  intInput   ,   // IN        - measurement
                     const int  MIN_RANGE_M ,  //     OUT   - male min range
                     const int  MAX_RANGE_M ,  //     OUT   - male max range
                     const int  MIN_RANGE_F ,  //     OUT   - female min range
                     const int  MAX_RANGE_F)   //     OUT   - female max range

{
    return ((sex  == 'm' || sex == 'M')
            && intInput >= MIN_RANGE_M && intInput <= MAX_RANGE_M) ||
            ((sex == 'f' || sex == 'F')
            && intInput >= MIN_RANGE_F && intInput <= MAX_RANGE_F);
}

```

```cpp
/*******************************************************************************
 * AUTHOR     : Blake Allard
 * STUDENT ID : 358888
 * ASN #6     : Military Academy
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/18/24
 *******************************************************************************/

#include "as6.h"       /* header file         */

/*******************************************************************************
 * EvaluateCandidate
 *   This function receives an acceptableHeight boolean variable & an
 *   acceptableWeight boolean variable, and returns whether or not both,
 *   one, or none of the variables are true or false based on each boolean
 *   variable's compound comparison statement, then the function outputs
 *   a string response dependent on which variables are true and/or false.
 *------------------------------------------------------------------------------
 *   ==> returns nothing - this will be used for determining a candidate's
 *       eligibility response dependent upon their character & integer
 *       inputs that are paired with their matching ranges.
 *******************************************************************************/

    void EvaluateCandidate(bool acceptableHeight ,  // IN & OUT  - eval height
                           bool acceptableWeight)   // IN & OUT  - eval weight


{
    if (acceptableHeight && acceptableWeight)
    {
        cout << "This candidate has been ACCEPTED!\n\n\n";
    }

    else if (acceptableWeight)
    {
        cout << "This candidate has been rejected based on the HEIGHT "
                "requirements.\n\n\n";
    }

    else if (acceptableHeight)
    {
        cout << "This candidate has been rejected based on the WEIGHT "
                "requirements.\n\n\n";
    }

    else
    {
        cout << "This candidate has been rejected based on the HEIGHT and "
                "WEIGHT requirements.\n\n\n";
    }
```

```
52 }
53
```