Homework #8 Solution
(C Programming for Beginners - OnLine)

Demo: The following code is a procedural way of writing code. It prints the weekly temperature, finds the min and max, and the average as well.

```
int weeklyTemp[] = { 69, 70, 71, 68, 66, 71, 70 };
int i, max = 0, min = 0;

// print temperatures
for (i = 0; i < 7; i++) {
      printf("\nThe temperature on day %d "
                                    "was %d: ", i + 1,
                                    weeklyTemp[i]);
}
printf("\n\n");

// find the max, min temperature
for (i = 0; i < 7; i++) {
      if (i == 0)
            max = min = weeklyTemp[i];
      if (weeklyTemp[i] > max)
            max = weeklyTemp[i];
      if (weeklyTemp[i] < min)
            min = weeklyTemp[i];
}
printf("The Minimum temperature is: %d\n", min);
printf("The Maximum temperature is: %d\n", max);

// get average
float total = 0, average;
for (i = 0; i < 7; i++)
      total += weeklyTemp[i];
average = total / 7;
printf("The average temperage for the week is: %5.2f ",
      average);
```

8.1 Now, modify the above code, which is all in one place, and break it into multiple functinos. Try to convert each piece of important code into a function.

a) Write a function called getTemperatures. Which asks the user to enter 7 temperatures for the week

Solution:

```
void getTemperatures(int temps[], int arraySize) {
	int i;

	for (i = 0; i < arraySize; i++) {
		printf("Enter the temp: ");
		scanf("%d", &temps[i]);
	}
	printf("\n\n");
}
```

b) Write a function called printTemperatures. Which prints the 7 temperatures for the week:

Solution:

```
void printTemperatures(int T[], int size) {
	int i;
	for (i = 0; i < size; i++) {
		printf("\nThe temperature on day %d was %d: ",
			i + 1, T[i]);
	}
	printf("\n\n");
}
```

c) Write a function called getMax, which returns the maximum temperature of the week.

Solution:

```
int getMax(int T[], int size) {

	int i, max = 0;
	// find the min temperature
	for (i = 0; i < 7; i++) {
		if (i == 0)
			max = T[i];
		if (T[i] > max)
			max = T[i];
	}
	return max;
}
```

d) Write a function called getMin, which returns the minimum temperature of the week.

Solution:

```
int getMin(int T[], int size) {

	int i, min = 0;
	// find the min temperature
	for (i = 0; i < 7; i++) {
		if (i == 0)
			min = T[i];
		if (T[i] < min)
			min = T[i];
	}
	return min;
}
```

e) Write a function called getAverage, which returns the average temperature of the week.

Solution:

```
float getAverage(int weeklyTemp[], int arraySize) {
	int i;
	float total = 0, average;
	for (i = 0; i < 7; i++)
		total += weeklyTemp[i];
	average = total / arraySize;

	return average;
}
```

f) Write a function called printStatistics that prints minimum, maximum and average of the week using above function

Solution:

```
void printStatistics(int T[], int size) {

	printf("The minimum of weekly temperature is: %d\n",
		getMin(T, size));
	printf("The maximum of weekly temperature is: %d\n",
		getMax(T, size));
	printf("The average of weekly temperature is: %5.2f\n",
		getAverage(T, size));
}
```

g) Write the code in main function to call all these functions

Solution:

```
int weeklyTemp[];

printf("\nCalling functions to do the work:\n\n");
getTemperatures(weeklyTemp, 7);
printTemperatures(weeklyTemp, 7);
printStatistics(weeklyTemp, 7); //in turn calls getMin,
                                //getMax, and getAverage;
```

Demo: Here is the solution for Homework# 4.6. It is written as a procedural program.

```
int counter1, counter2;
char hChar1, vChar1;
int  ht1, wd1;
char answer = 'y';

printf("\nUsing for-loop and user values, continuously:
       \n");

while (answer == 'y') {
      printf("\nPlease enter height of a box: ");
      scanf("%d", &ht1);
      printf("\nPlease enter width of a box: ");
      scanf("%d", &wd1);

      //you first need to flush the buffer, which still has
      //'\n' character due to pressing enter
      while (getchar() != '\n');

      printf("\nPlease enter the vertical charcters to draw
             box: ");
      scanf("%c", &vChar1);

      //don't forget to remove the newline character after
      //reading just a charcter in above code
      while (getchar() != '\n');

      printf("\nPlease enter the horizontal charcters to
            draw box: ");
      scanf("%c", &hChar1);

      printf("\n");
      for (counter1 = 1; counter1 <= wd1; counter1++)
      {
            printf("%c", hChar1);
      }
      printf("\n");
```

```
                for (counter1 = 1; counter1 <= ht1 - 2; counter1++)
                {
                      printf("%c", vChar1);
                      for (counter2 = 1; counter2 <= wd1 - 2;
                                                      counter2++)
                            printf(" ");
                      printf("%c\n", vChar1);
                }

                for (counter1 = 1; counter1 <= wd1; counter1++)
                {
                      printf("%c", hChar1);
                }
                printf("\n");

                //clean up the newline after last character read
                while (getchar() != '\n');

                printf("Continue? Type 'y' for yes: ");
                scanf("%c", &answer);

          }
```

8.2 Now, modify and break the above code into four functions:

a) A function, drawHorizontalLine, which draws horizontal lines

"----------"

Solution:

```
static void drawHorizontalLines(int wd1, char hChar1) {
      int x;
      for (x = 1; x <= wd1; x++) {
            printf("%c", hChar1);
      }
      printf("\n");
}
```

b) A function, drawVerticalLine, which draws vertical lines

"| |"
"| |"
"| |"

Solution:

```
static void drawVerticalLines(int wd1, int ht1, char vChar1) {
	int x, y;
	for (x = 1; x <= ht1 - 2; x++)
	{
		printf("%c",vChar1);
		for (y = 1; y <= wd1 - 2; y++)
			printf(" ");
		printf("%c\n", vChar1);
	}
}
```

c) A function, drawBox, which calls the drawHorizontalLine, and drawVerticalLine to draw the box

Solution:

```
static void drawBox(int wd1, int ht1, char hChar1, char vChar1) {
	drawHorizontalLines(wd1, hChar1);
	drawVerticalLines(wd1, ht1, vChar1);
	drawHorizontalLines(wd1, hChar1);
	printf("\n\n");
}
```

d) Call drawBox from main function

Solution:

```
		int counter1, counter2;
		char hChar1, vChar1;
		int  ht1, wd1;
		char answer = 'y';

		printf("\nUsing for-loop and user values, continuously: \n");

		while (answer == 'y') {
			printf("\nPlease enter height of a box: ");
			scanf("%d", &ht1);
			printf("\nPlease enter width of a box: ");
			scanf("%d", &wd1);

			//you first need to flush the buffer, which still has
			//'\n' character due to pressing enter
			while (getchar() != '\n');

			printf(
				"\nPlease enter the vertical charcters to draw box: ");
			scanf("%c", &vChar1);

			//don't forget to remove the newline character after
			//reading just a charcter in above code
			while (getchar() != '\n');

			printf(
				"\nPlease enter the horizontal charcters to draw box: ");
			scanf("%c", &hChar1);
```

```
                printf("\n");

                drawBox(wd1, ht1, hChar1, vChar1);

                //clean up the newline after last character read
                while (getchar() != '\n');

                printf("Continue? Type 'y' for yes: ");
                scanf("%c", &answer);

        }

printf("\nThank you for using my program\n");
```