Module Title

Principles of Programming

Weekly Assignment - Practical 1

Year

2025

Student Name: NIRVIK K.C.

UWE ID: 25024649

Assignment Submission Date: December 19, 2025

This assignment consists of the programming questions from practical exercises related to the topics from week 1.

Questions

1. Type the following code in the file named HelloWorld.c and display it.

   Given code :

   Type the following code in HelloWorld.c:

   /* Print a message on the screen */

   #include int main() {

   /* code */

   printf("Hello World! \n");

   return 0;

   }

Answer:

Following code for input:

/ * Print monthly expenditure Part 1, Part a */

/ * FIrst question - Type the following code in HelloWorld.c */

```
#include <stdio.h>
  int main()
 {
   printf("Hello World! \n");
 }

/* Output : Hello World! will be displayed in the terminal. */
```

Output obtained in execution:

2. (a) Type the following code into your program file named MonthlyExpenditure.c and display it:

Given code :

```
/* Print monthly expenditure
Practical 1, Part 2 (a)
@author your name */

#include<stdio.h>
int main() {
/* code */
return 0;
}
```

(a). Type the following code into the sample program to replace /* code */:

```
//define variables and assign values to them
float foodExpenses = 300.0; //variable for food expenses
float leisureExpenses = 100.0; //assign 100.0 to leisureExpenses
float clothesExpenses = 50.0;
float totalSpent; //variable for total expenses
totalSpent = foodExpenses + leisureExpenses + clothesExpenses;
printf("The total expenditure this month was £%.2f\n\n", totalSpent);
```

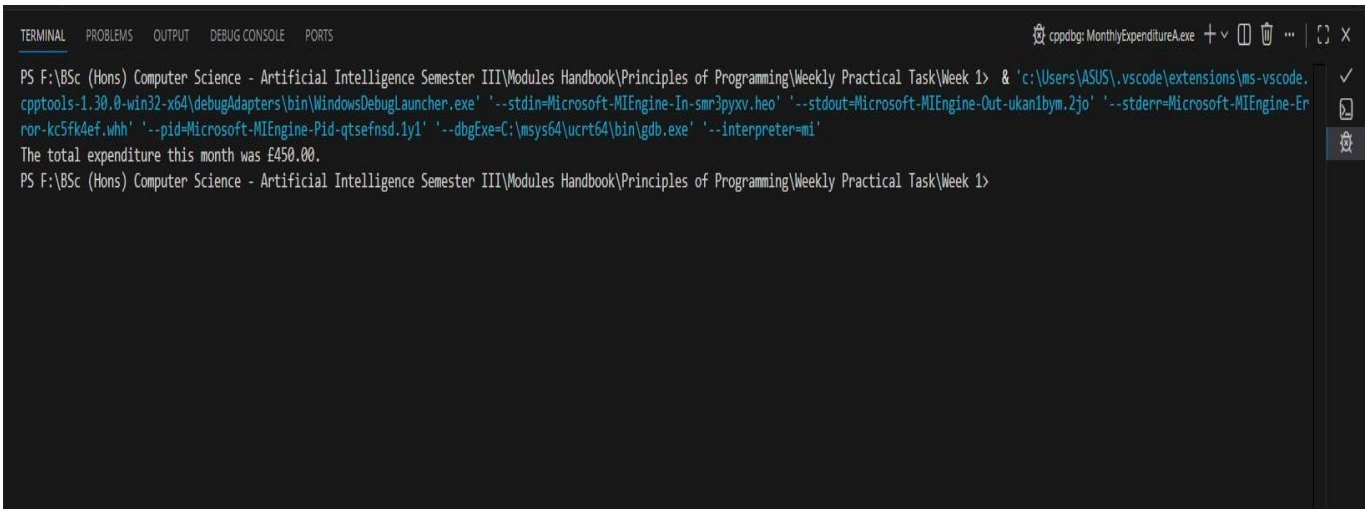
Answer:

Following code for input:

```
/* Print monthly expenditure
Practical 1, Part 2 (a)
Nirvik K.C. */

// Type the following code in MonthlyExpenditureA.
// Code:
/*  define variables and assign values to them
   float foodExpenses = 300.0; //variable for food expenses
   float leisureExpenses = 100.0; //assign 100.0 to leisureExpenses
   float clothesExpenses = 50.0;
   float totalSpent; //variable for total expenses
   totalSpent = foodExpenses + leisureExpenses + clothesExpenses;
   printf("The total expenditure this month was £%.2f\n\n", totalSpent); */

#include <stdio.h>
int main()
{
   float foodExpenses = 300.00;
   float leisureExpenses = 100.00;
   float clothesExpenses = 50.00;
   float totalSpent;
```

```
    totalSpent = foodExpenses + leisureExpenses + clothesExpenses;
    printf("The total expenditure this month was £%.2f.\n", totalSpent);
    return 0;
}
// Output : The total expenditure this month was £450.00.
```

Output obtained in execution:



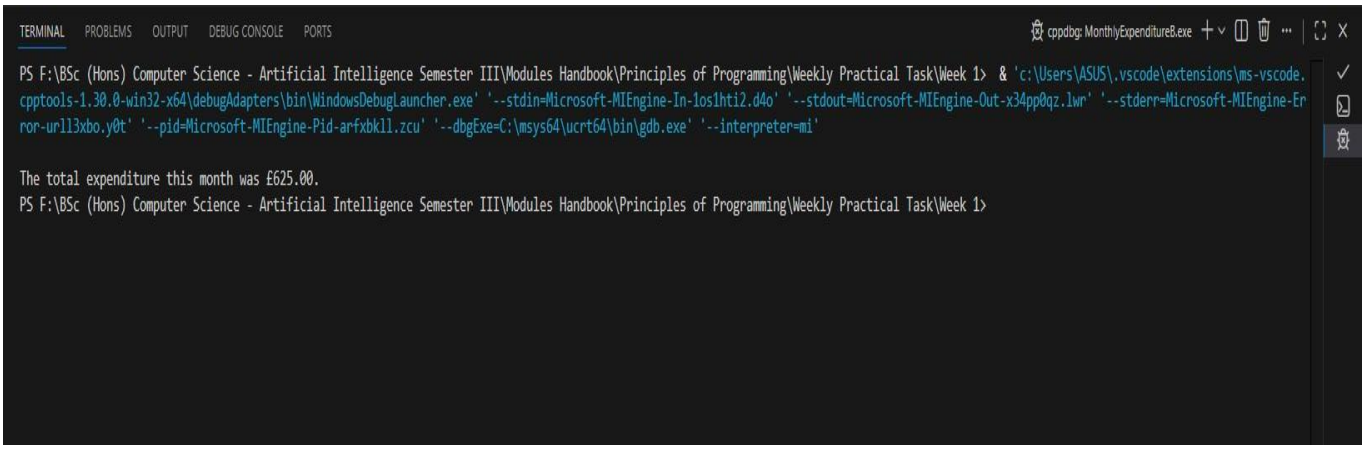
2. (b) Save MonthlyExpenditureA.c as MonthlyExpenditureB.c for this question. Edit MonthlyExpenditureB.c to calculate your total money spent last month on food, leisure and clothes (i.e. to replace those default values with your own expenses). Run the edited program.

Answer:

Following code for input:

```
/* Print monthly expenditure
Practical 1, Part 2 (b)
Nirvik K.C. */
/* Edit the program MonthlyExpenditureA to MonthlyExpenditureB to calculate
the total money spent last month on food, leisure, and clothes.
Also, replace those default values from the previous program with your own. */

#include <stdio.h>
int main()
{
   float foodExpenses = 350.00;
   float leisureExpenses = 175.00;
   float clothesExpenses = 100.00;
   float totalSpent;
   totalSpent = foodExpenses + leisureExpenses + clothesExpenses;
   printf("\n The total expenditure this month was £%.2f.\n", totalSpent);
   return 0;
}
// Output : The total expenditure this month was £625.00.
```

Output obtained in execution:



2. (c) Save MonthlyExpenditureB.c as MonthlyExpenditureC.c Extend the program so that it can be used to calculate your total money spent last month on food, leisure, clothes, accommodation and travel. Run the extended program.

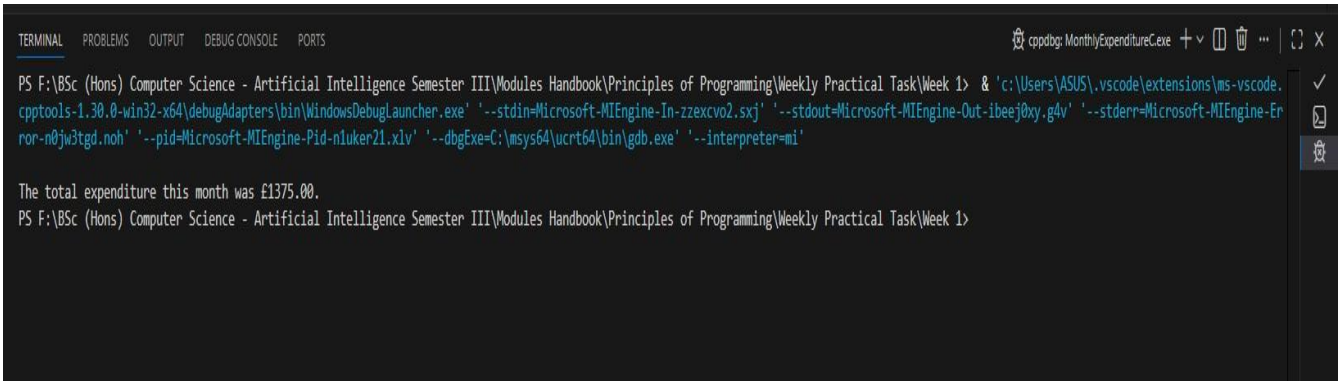
Answer:

Following code for input:

```
/* Print monthly expenditure
Practical 1, Part 2 (c)
Nirvik K.C. */
/* Edit, extend and save the MonthlyExpenditureB program to
MonthlyExpenditureC
program to calculate the total money spent last month on food, leisure, clothes,
accommodation, and travel.
Also, add default values for the new variables (accommodation and travel). */
```

```
#include <stdio.h>
int main()
{
    float foodExpenses = 350.0;
    float leisureExpenses = 175.00;
    float clothesExpenses = 100.00;
    float accommodationExpenses = 600.00;
    float travelExpenses = 150.00;
    float totalSpent;
    // Calculate the total expenses
    totalSpent = foodExpenses + leisureExpenses + clothesExpenses +
accommodationExpenses + travelExpenses;
    printf("\nThe total expenditure this month was £%.2f.\n", totalSpent);
    return 0;
}
// Output : The total expenditure this month was £1375.00.
```

Output obtained in execution:



2. (d) Save MonthlyExpenditureC.c as MonthlyExpenditureD.c. Further extend the program in so that it can be used to take input from the keyboard, calculate your total money spent last month food, leisure, clothes, accommodation and travel, and display your total money on the screen. Run the further extended program.

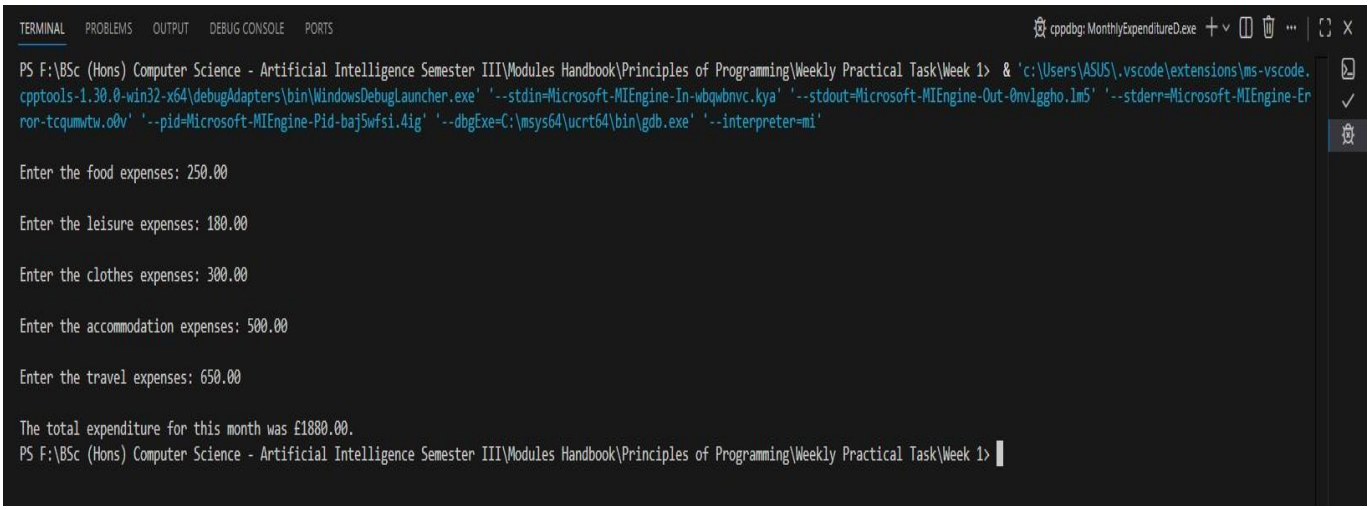
Answer:

Following code for input:

```
/* Print monthly expenditure
Practical 1, Part 2 (c)
Nirvik K.C. */
/* Edit, extend and save the MonthlyExpenditureC to MonthlyExpenditureD
program to take input from the keyboard, calculate total
money spent last month food, leisure, clothes, accommodation and travel, and
display your total money on the screen. */
#include <stdio.h>
int main()
{
   float foodExpenses;
   float leisureExpenses;
   float clothesExpenses;
   float accommodationExpenses;
   float travelExpenses;
   float totalSpent;
```

```
    // To take input from the user
    printf("\nEnter the food expenses: ");
    scanf("%f", &foodExpenses);
    printf("\nEnter the leisure expenses: ");
    scanf("%f", &leisureExpenses);
    printf("\nEnter the clothes expenses: ");
    scanf("%f", &clothesExpenses);
    printf("\nEnter the accommodation expenses: ");
    scanf("%f", &accommodationExpenses);
    printf("\nEnter the travel expenses: ");
    scanf("%f", &travelExpenses);

    // Calculate the total expenses
    totalSpent = foodExpenses + leisureExpenses + clothesExpenses +
accommodationExpenses + travelExpenses;

    // Display the total expenses in the terminal
    printf("\nThe total expenditure for this month was £%.2f.\n", totalSpent);
    return 0;
}
// Output : The total expenditure for this month was £1880.00.
```

Output obtained in execution:



2. (e) Save MonthlyExpenditureD.c as MonthlyExpenditureE.c. Assume that you spend a fixed amount on your accommodation expenses each month. Change your extended program in (d) so that your accommodation expenses are represented as a constant. Run your final program.

Answer:

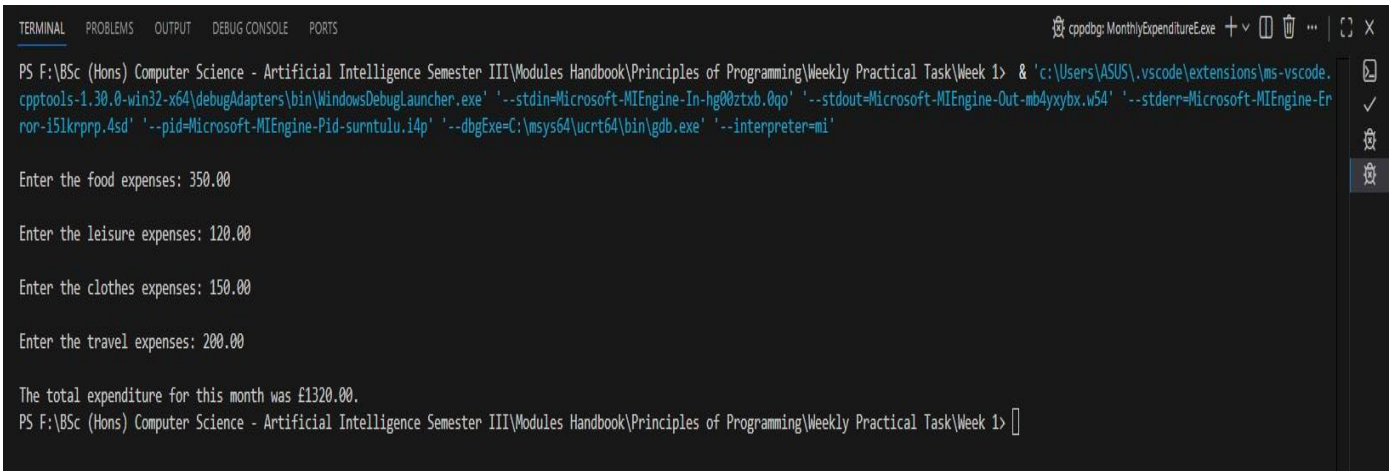
Following code for input:

```
/* Print monthly expenditure

Practical 1, Part 2 (e)

Nirvik K.C. */
```

```
/* Edit and save the MonthlyExpenditureD.c to MonthlyExpenditureE.c to spend a
fixed amount on the accommodation expenses each month.

Make the required change in the program so that the accommodation expenses are
represented as a constant. */

#include <stdio.h>

int main()

{

   float foodExpenses;

   float leisureExpenses;

   float clothesExpenses;

   float travelExpenses;

   // Define accommodation expense as a constant

   const float ACCOMMODATION_EXPENSES = 500.00;

   float totalSpent;

// To take input from the user
```

```
    printf("\nEnter the food expenses: ");
    scanf("%f", &foodExpenses);
    printf("\nEnter the leisure expenses: ");
    scanf("%f", &leisureExpenses);
    printf("\nEnter the clothes expenses: ");
    scanf("%f", &clothesExpenses);
    // Accommodation expenses are fixed so no input needed
    printf("\nEnter the travel expenses: ");
    scanf("%f", &travelExpenses);
    // Calculate the total expenses
    totalSpent = foodExpenses + leisureExpenses + clothesExpenses +
ACCOMMODATION_EXPENSES + travelExpenses;
    // Display the total expenses in the terminal
    printf("\nThe total expenditure for this month was £%.2f.\n", totalSpent);
    return 0;
} // Output : The total expenditure for this month was £1320.00.
```

Output obtained in execution:



2. (f) Write a summary of all the programming concepts that we have covered in this practical. Note that we have now a list of 5 programs for monthly expenditures in the practical1 folder. It is a good practice to create a new version of a program for a different part of an exercise (whenever we have extended/changed an existing program so we will keep the old version too).

Answer:

The practical exercises from week 1 helped me understand the basic fundamentals for programming in C language. In the practical, we developed five programs, named MonthlyExpenditureA.c to MonthlyExpenditureE.c. These programs focused on calculating the total amount spent on variety of monthly expenses such as food, leisure, clothes, accommodation, and travel. Throughout the learning process, we applied various concepts of the C language.

Firstly, we explored the basic structure of C, where we incorporated necessary components needed in the program, such as include directives, the int main() function, variable declarations, data type, executable statements, and the use of return (0) statement. For variables, we focused on the declaration and utilization of float variables to store decimal amounts for representing expenses. This also helped us to understand the arithmetic calculations to find the total expenditure, where the (+) operator was used to calculate the result. Another thing to remember was the way to print formatted output using the printf() function, with the use of format specifier like %.2f to display decimal values such as £1320.00.

Function scanf() helped us to understand user input handling, where we learned about the address operator (&) and its use for reading the values directly from the user with the help of the keyboard. We learned that values can be defined as fixed using the const keyword and use of uppercase letters can help us to identify them. We learned about using single line comments (//) and multi-line(/* */) comments in the program for clarity and clear understanding. We learn about code modifications where we extended our programs by adding features, changing user inputs, altering values, and using the const keyword.

Overall, the practical exercises from week 1 emphasized on the step-by-step approach towards solving the problems and covered the essential fundamentals for C programming.

Part 3 - More Programming exercises

3. (a) (Population projection) The U.S. Census Bureau projects population based on the following assumptions:
   • One birth every 7 seconds
   • One death every 13 seconds
   • One new immigrant every 45 seconds
   Write a program to display the population for each of the next five years. Assume the current population is 312,032,486 and one year has 365 days.
   Hint: In Java, if two integers perform division, the result is an integer. The fractional part is truncated. For example, 5 / 4 is 1 (not 1.25) and 10 / 4 is 2 (not 2.5). To get an accurate result with the fractional part, one of the values involved in the division must be a number with a decimal point. For example, 5.0 / 4 is 1.25 and 10 / 4.0 is 2.5.

Answer:

Following code for input:

```
/* Print the total population for next five years
Practical 1, Part 3 (a)
Nirvik K.C. */

/* Write a program to display the population for each of the next five years.
Assume the current population is 312,032,486 and one year has 365 days. */
```

```
#include <stdio.h>
int main()
{
   double population = 312032486.0;
   double time = 365 * 24 * 60 * 60;
   float birth = time / 7;      // One birth every 7 seconds
   float death = time / 13;     // One death every 13 seconds
   float immigrant = time / 45; // One new immigrant every 45 seconds

   double net_population = birth - death + immigrant;
   printf("\nYearly Population: \n");

   double population_year1 = population + net_population;
   printf("First year population: %.0lf\n", population_year1);

   double population_year2 = population_year1 + net_population;
   printf("Second year population: %.0lf\n", population_year2);

   double population_year3 = population_year2 + net_population;
   printf("Third year population: %.0lf\n", population_year3);

   double population_year4 = population_year3 + net_population;
   printf("Fourth year population: %.0lf\n", population_year4);
```

```
    double population_year5 = population_year4 + net_population;
    printf("Fifth year population: %.0lf\n", population_year5);
   return 0;
}

/* Output :
Yearly Population:
First year population: 314812583
Second year population: 317592680
Third year population: 320372776
Fourth year population: 323152873
Fifth year population: 325932970
 */
```

Output obtained in execution:

3. (b) (Convert Celsius to Fahrenheit) Write a program that reads a Celsius degree in a double value from the console, then converts it to Fahrenheit and displays the result. The formula for the conversion is as follows: Fahrenheit = (9 / 5) * Celsius + 32.
   Hint: In Java, 9 / 5 is 1, but 9.0 / 5 is 1.8.

Answer:

Following code for input:

```
/* Conversion from Celsius to Fahrenheit
Practical 1, Part 3 (b)
Nirvik K.C. */

/* Write a program that reads a Celsius degree in a double
value from the console, then converts it to Fahrenheit and displays the result.
The formula for the conversion is as follows: Fahrenheit = (9 / 5) * Celsius + 32.
*/
#include <stdio.h>
int main()
{
   double celsius;
   double fahrenheit;
```

```
// To take input - temperature in Celsius from the user
    printf("\nEnter a degree in Celsius: ");
    scanf("%lf", &celsius);
    // Convert Celsius to Fahrenheit using the given formula
    fahrenheit = (9.0 / 5) * celsius + 32.0;
    printf("\n%.0lf degrees Celsius is %1.lf Fahrenheit.\n", celsius, fahrenheit);
    return 0;
}
/* Output:
Enter a degree in Celsius: 50

50 degrees Celsius is 122 Fahrenheit.
 */
```

Output obtained in execution:

3. (c) Write a program that prompts the user to enter a monthly saving amount and displays the account value after the sixth month.

   Given example:

   Suppose you save $100 each month into a savings account with the annual interest rate 5%. Thus, the monthly interest rate is 0.05/12 = 0.00417. After the first month, the value in the account becomes

   100 * (1 + 0.00417) = 100.417

   After the second month, the value in the account becomes

   (100 + 100.417) * (1 + 0.00417) = 201.252

   After the third month, the value in the account becomes (100 + 201.252) * (1 + 0.00417) = 302.507 and so on.

Answer:

Following code for input:

```
/* Calculate account value after the certain time
Practical 1, Part 3 (c)
Nirvik K.C. */
/* Write a program that prompts the user to enter a monthly savings amount and displays the
account value after the sixth month. */
/* Account value compounded monthly at an annual interest rate of 5%. */
```

```
#include <stdio.h>
int main()
{
   float monthly_saving;
   float interest_rate = 0.05;
   float monthly_interest_rate = interest_rate / 12;

   // To take input - monthly saving value from the user.
   printf("\nEnter the monthly saving amount: ");
   scanf("%f", &monthly_saving);
   float month1 = monthly_saving * (1 + monthly_interest_rate);
   float month2 = (monthly_saving + month1) * (1 + monthly_interest_rate);
   float month3 = (monthly_saving + month2) * (1 + monthly_interest_rate);
   float month4 = (monthly_saving + month3) * (1 + monthly_interest_rate);
   float month5 = (monthly_saving + month4) * (1 + monthly_interest_rate);
   float month6 = (monthly_saving + month5) * (1 + monthly_interest_rate);

   // Display the output
   printf("\nAccount value after the sixth month: $%.2f.\n", month6);
   return 0;
}
```

```
/*
Output :
Enter the monthly saving amount: 100

Account value after the sixth month: $608.81.
*/
```

Output obtained in execution: