Unit 1 Programming Problems Worksheet

Programming Problem 1 – Computer Configuration

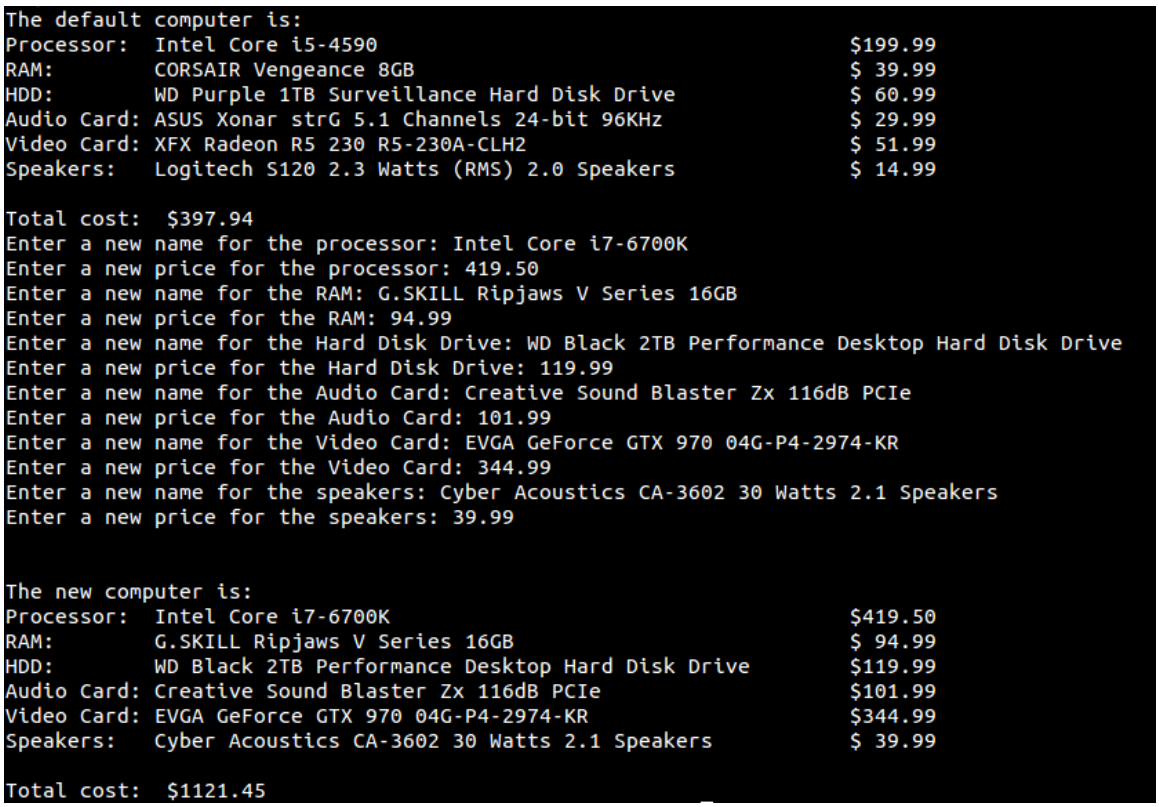
Using the web sites of various computer vendors of your choice, pick a computer that would satisfy the needs of an average college student, and collect all characteristics of the computer that you consider interesting, including the price. Write a program that displays the defining characteristics of the computer in a format that would attract potential users.

Continue your exercise by simulating an upgrade for the computer chosen earlier. Choose a better processor, more RAM, a different hard disk drive, an audio card, a video card, and speakers. Find prices for all the components mentioned above. Continue your program, defining appropriate variables for the new components, and the price of the new, upgraded computer. Read all prices from the user, and calculate the price of the new computer. Display the new configuration, with all the characteristics defined earlier, as well as the price.

Grading Rubric

Task	Points
Initial Computer Configuration	3
Computer upgrade - choice of components	2
Written program	5
Overall usage of best practices in programming	2
Total	**12**

Screenshots



Programming Problem 2 – Calculating a Number

As part of the discussion forum for this unit, you will be solving a programming problem from the book. This assignment will have you calculating the chapter and the number of the programming problem that you will solve for the online discussion forum.

To calculate the chapter from which you solve the programming exercise:

- Divide the integer number representing your student ID by 4, consider the remainder and increment it by 2. The result you obtain represents the chapter number, and it should be either 2, 3 ,4 or 5.

Depending on the chapter number obtained above, consider the following rules in calculating the problem number to solve:

- If the chapter number is 2, divide your student ID by 26, consider the remainder and increment it by 1. The result you obtain represents the number of the programming exercise you will solve for online discussions, which should be from chapter 2.

- If the chapter number is 3, divide your student ID by 34, consider the remainder and increment it by 1. The result you obtain represents the number of the programming exercise you will solve for online discussions, which should be from chapter 3.

- If the chapter number is 4, divide your student ID by 46, consider the remainder and increment it by 1. The result you obtain represents the number of the programming exercise you will solve for online discussions, which should be from chapter 4.

- If the chapter number is 5, divide your student ID by 38, consider the remainder and increment it by 1. The result you obtain represents the number of the programming exercise you will solve for online discussions, which should be from chapter 5.

After calculating the number of the chapter, and the number of the programming exercise to solve, ask the user to enter the page number where the specific problem is located in the textbook. Display the requirement for the programming exercise using the following format:

> *"Please solve programming exercise … (include here the number of the exercise) from chapter … (include here the number of the chapter), from page … (include here the page number)."*

Grading Rubric

Task	Points
Working solution	6
Overall usage of programming best practices	2
Total	8

Screenshots

```
Enter student id: 1234567
Enter page of problem 24 from chapter 5: 195
Please solve programming exercise 24 from chapter 5, from page 195.
```

Programming Problem 3 – Vending Machine

Write a program that simulates the functionality of a vending machine having the following characteristics:

- The vending machine offers 5 products

- The vending machine accepts coins, 1 dollar bills, and 5 dollar bills

- The change is always given in coins, with maximum possible number of coins in each value: 25, 10, 5 or 1 cent.
- The selections available for user are numbers from 1 to 5.
- The user enters the money – simulate the action through a loop that ends when the user enters 0. Each coin, or paper bill will be read individually.
- The user makes the selection, and the machine allows a maximum 4 other selections if the amount entered doesn't cover the price of the item.
- Once an item is delivered, the machine gives the change in coins.
- There is no increment for the money during one selection.
- The user can stop the selection at any time by entering 0 for the product selection.
- If the user chooses to cancel the selection, the machine returns the initial amount in coins.
- Display the outcome of the operation for each alternative you consider possible for the vending machine.
- Make sure that the machine returns the correct change at all times.

Use appropriate selection and repetition loops to solve the problem.

Write a test program that would take at least 3 combinations of amounts entered and choices of products, and displays the results of all three trials. You can include the vending machine program as a method in the testing program.

Grading Rubric

Task	Points
Correct program written according to the requirements listed above	10
Correct testing program including three different data combinations	4
Overall usage of best practices in programming	2
Total	**16**

Screenshots

```
Prices:
01: Soda                - $0.75
02: Water               - $0.32
03: Candy Bar           - $0.55
04: Trail Mix           - $1.55
05: Chips               - $1.00
06: Crackers            - $0.81
Put money in machine: 1
Put money in machine: 9
Vending Machine cannot accept this money
Put money in machine: .25
Put money in machine: .6
Vending Machine cannot accept this money
Put money in machine: .05
Put money in machine: 0
Enter id of product: 1
Product vended
Change
Quarters: 2
Dimes:    0
Nickles:  1
Pennies:  0
Prices:
01: Soda                - $0.75
02: Water               - $0.32
03: Candy Bar           - $0.55
04: Trail Mix           - $1.55
05: Chips               - $1.00
06: Crackers            - $0.81
Put money in machine: 1
Put money in machine: 1
Put money in machine: 0
Enter id of product: 1
Product vended
Change
Quarters: 5
Dimes:    0
Nickles:  0
Pennies:  0
```

```
Prices:
01: Soda                - $0.75
02: Water               - $0.32
03: Candy Bar           - $0.55
04: Trail Mix           - $1.55
05: Chips               - $1.00
06: Crackers            - $0.81
Put money in machine: 1
Put money in machine: 0
Enter id of product: 2
Product vended
Change
Quarters: 2
Dimes:    1
Nickles:  1
Pennies:  3
```

Programming Problem 4 – Broken GPS

Consider yourself driving with 60 miles/hour in a city that has only grid like streets, and your GPS is broken.

The specifications of the problem are:

- With (x, y) as the coordinates of the car, consider the initial position (0,0).
- At each intersection, the only directions available are North, South, East or West
- The GPS is broken, and it chooses the direction randomly at each intersection, which happens every 5 minutes
- The assumption is that the car has the same speed at all times, including when it changes direction and turns.

Write a program that calculates the distance from the initial point to the location point of the driver after one hour of driving.

Grading Rubric

Task	Points
Working solution for the problem	10
Overall usage of programming best practices	2
Total	**12**

Screenshots

```
az@ASUS-K55A:~/Dropbox/School/CS/CSC_201/01/Problem_04$ for i in {0..8} ; do java Main ;done
Car endded at: (0, 2), distance: 10.000000 miles
Car endded at: (1, 1), distance: 7.071068 miles
Car endded at: (1, 3), distance: 15.811388 miles
Car endded at: (-4, -2), distance: 22.360680 miles
Car endded at: (-6, 0), distance: 30.000000 miles
Car endded at: (-2, -4), distance: 22.360680 miles
Car endded at: (-2, -2), distance: 14.142136 miles
Car endded at: (-5, 1), distance: 25.495098 miles
Car endded at: (1, -1), distance: 7.071068 miles
```

Programming Problem 5 – Complex Numbers

A complex number is defined as z=a+i*b, where a is the real part, and b is the imaginary part. In other words, in order to define a complex number, we need the two floating numbers a and b.

Write methods that perform for each of the following operations with complex numbers z1 = a1 + i*b1, and z2 = a2 + i*b2:

- Addition: z1 + z2=(a1+a2) + i*(b1+b2)
- Subtraction: z1 - z2=(a1-a2) + i*(b1-b2)
- Multiplication: z1*z2 = (a1*a2 – b1*b2) + i*(a1*b2 + b1*a2)
- Division: z1/z2 = (a1*a2 +b1*b2)/(a2^2 + b2^2) + i*(b1*a2 – a1*b2)/(a2^2 + b2^2)

Create a test program that asks for the real and imaginary parts of two complex numbers from the user, and displays the results of the four operations, writing the formula as shown above, and replacing the a1, a2, b1 and b2 with the numbers entered by the user.

Grading Rubric

Task	Points
Working solution for the problem	10
Overall usage of programming best practices	2
Total	**12**

Screenshots

```
Enter z1 real part: 7
Enter z1 imaginary part: 5
Enter z2 real part: 6
Enter z2 imaginary part: 9.8

7.0+i*5.0 + 6.0+i*9.8 = 13.0+i*14.8
7.0+i*5.0 - 6.0+i*9.8 = 1.0+i*-4.800000000000001
7.0+i*5.0 * 6.0+i*9.8 = -7.0+i*98.60000000000001
7.0+i*5.0 / 6.0+i*9.8 = 0.5877034358047015+i*-0.29233565586186006
```