

Foundations of Programming

Individual Assignment

DUE : Refer to timetable

Question 1 (10 marks)

You are to write a C program that accepts three **integer** values from the user representing the three sides of a triangle. Your program should report to the user if it is a valid triangle and what sort of a triangle: equilateral, scalene or isosceles. (Hint: a triangle is only valid if the sum of any given pair of sides is greater than the remaining side.) Your solution does not need to be modular and so a high-level algorithm is not required, although you are welcome to include one if you wish.

Question 2 (20 marks)

You should read this question as if the change giver is a bank teller who has to give amounts of money to customers but can't count very well so needs your program to help them.

You are asked to write a simple C program that will accept an **integer** value in the range of 5-95 and as a multiple of 5 representing the number of cents to give to a customer in their change. The program should calculate how many coins of each denomination and display this to the user. Valid coin values are 50, 20, 10 and 5. Your solution (program and algorithm) should be modular in nature. This requires the submission of a high-level algorithm and suitable decompositions of each step.

Note that for this problem the principle of code reuse is particularly important and a significant number of marks are allocated to this. You should attempt to design your solution such that it consists of a relatively small number of functions that are as general in design as possible and you should have one function in particular that can be reused (called repeatedly) in order to solve the majority of the problem. If you find that you have developed a large number of functions which each performs a similar task (or have a lot of repeated code) then attempt to analyse your design to generalise the logic so that it may be reused.

Question 3 (20 marks)

You are to extend your solution to Question 2 above by allowing the user to enter the amount of change as dollars and cents as a **double** (real number). Valid dollar values are 100, 50, 20, 10, 5, 2 and 1. Your program should output the correct combination of dollar and cent denominations required and should be able to work **for any valid dollar amounts**. Cents amounts may be should be multiples of 5 as in the previous question.

In this question you will need to explicitly convert a double to an integer to break the inputted double down to its component parts as integers. You can use the following code as the basis for doing this:

```
float change;  
int dollar, cents;
```

...

```
dollar = (int) change;  
cents = (int) (((change - dollar)*100) + 0.5);
```

Again your solution for this problem should be modular.

Assignment Submission

1. Write your name clearly on the cover page using the Assignment Cover Form.
2. Your assignment should be type-written. (Submit both hard copy and soft copy. Soft copy to be submitted on a CD. Please label the CD with your name and module code).
3. There will be **penalty** for plagiarism.
4. It is advisable to keep a soft copy of your submitted assignment in case of unforeseen circumstances that requires a re-submission of your assignment

Grading Criteria

Question 1

	Marks
Program executed without runtime error	5
Adopt good programming practices <ul style="list-style-type: none">• Meaningful variable names• No redundant code• Modula programming	3
Comprehensive test cases	1
Documentation	1
Total	10

Question 2

	Marks
Program executed without runtime error	10
Adopt good programming practices <ul style="list-style-type: none"> • Meaningful variable names • No redundant code • Modula programming 	5
Comprehensive test cases	2
Documentation	3
Total	20

Question 3

	Marks
Program executed without runtime error	10
Adopt good programming practices <ul style="list-style-type: none"> • Meaningful variable names • No redundant code • Modula programming 	5
Comprehensive test cases	2
Documentation	3
Total	20