CS 101: Computer Programming and Utilization

22-Revision

Instructor: Sridhar Iyer
IIT Bombay

Tutorial problem 1 – Parameter Passing

```
struct tuple { double x, y; };

double f (tuple &f1, tuple *f2, tuple f3) {

double res = (f1.x – f2→x) * (f3.y – f2→y);

f1.x = 20;  f2→x = 30;  f3.x = 40;

return res;

}

int main() {

tuple p = {0,0}, q = {3,4}, r = {4,3};  double s = f (p, &q, r);

cout << s << p.x << q.x << r.x << endl;

} // What will be the output of this program? Why?
```

Run: demo22-param.cpp

Tutorial problem 2 – Matrix Functions

1. Write a function '*evenM*" that:
   - takes a 3x3 matrix of integers as input
   - returns true if all the numbers in the matrix are even, else returns false.
2. Write a program that:
   - takes a 9x9 matrix of integers, M, as input
   - Uses *evenM* to find all the 'even' sub-matrices in M.

Compare with - demo22-matrix.cpp

Tutorial problem 3 – String Functions

Write a function 'isSubstring' that:

- takes two strings as input
- returns true if the second string is contained in the first string, else returns false.

For example, isSubstring("endsem", "dse") returns true; isSubstring("endsem", "dem") returns false.

Compare with - demo22-strings.cpp

Tutorial problem 4 – Recursion and Sorting

Recall the sorting algorithms in demo14-sorting.cpp

Write a recursive version for one or more of:

- InsertionSort()
- SelectionSort()
- BubbleSort()

Solution and further practice problems:

- www.codextream.com/write-a-c-program-or-algorithm-for-insertion-sort-using-recursion/

Tutorial problem 5 – Linked List

1. Write a function 'elimDup' that takes an array A, of integers, and eliminates all the duplicate entries in it.

- The function should take A and its length as input, modify A so that non-duplicate entries are contiguous from A[0], and give back the modified A and its length.

2. Write a program to eliminate duplicate entries from a linked list, without using an array.

- A trivial solution is to copy values from the list into an array, call elimdup() on the array, and recreate the list from the modified array.

Compare with - demo22-linklist.cpp

Tutorial problem 6 – File Handling

Write a program to open a file called “xyz.txt” and do the following:

- Print the size of the file in bytes.
- Count the number of words in the file.

You may assume that the file contains only characters and adjacent words are separated by exactly one blank-space. Also assume that each blank-space and end-of-line symbol is one byte.

Compare with - demo22-files.cpp

Tutorial problem 7 - Classes

Define a class *rational* that:

- Can store a rational number of the form a/b, where a and b are integers.
- Has constructors to create rational numbers.
  - You may need a helper function to reduce a rational number to its lowest form.
- Provides functions for Addition, Subtraction, Multiplication and Division operations on rational numbers.

Compare with - demo22-rational.cpp

Tutorial problem 8 - Inheritance

Predict the output at each step of

- demo22-accessCtrl.cpp

Tutorial problem 9 – Real life context

A bank offers two types of accounts to its customers: Savings Account and Current Account. While both accounts permit deposits and withdrawals, they follow different rules:

When you deposit an amount into the Current Account, it is simply added to the account balance. There is no interest calculation. When you deposit an amount into the Savings Account, 2% of the account balance (after the deposit) is added to the balance, as interest.

When you withdraw from the Current Account, if the account balance (after withdrawal) falls below 500, a penalty of 2% of the balance is further debited from the account. When you withdraw from the Savings Account, there is no such penalty.

Tutorial problem 9 – contd.

```
class account { // This base class is given to you
protected:
  string name;  int  accNo; char accType; float balance;
public:
   void input_accinfo(); // Input Customer name, A/c number, Type
   void display_accinfo(); // Output above information
   void show_balance(); // Output the balance
};
```

Define two derived classes SavingsAcc and CurrentAcc, which use the base class and provide additional functions – deposit and withdraw – as described earlier.

Compare with: demo22-accounts.cpp

End of cs101 classes

We have had a long journey,

- Statements – Conditions – Loops – Variables –Threads - Arrays – Matrices – Pointers – Functions – Recursion – Sorting - Strings – Structures – Linked Lists - Classes – Inheritance – File Handling

- In addition to C++, you have also learnt Scratch
  - Scratch programming is not included for endsem exam.

You will learn more in subsequent courses.

I hope all of you enjoyed some part of the course,

- either in terms what you learned (topics) or in terms of how you learned (Think-Pair-Share activities).

Do write your comments in the course evaluation.