# Practice Guide for Programming Practices

Education and Research

Infosys®
POWERED BY INTELLECT
DRIVEN BY VALUES



| | |
|---|---|
| **Author(s)** | Anooja Mary Jacob |
| **Authorized by** | Srikantan Moorthy |
| **Creation/Revision Date** | Jun - 2009 |
| **Version** | 1.0 |

Education and Research Department
Infosys Technologies Limited
Electronics City
Hosur Road
Bangalore - 561 229, India.

Tel: 91 80 852 0261-270
Fax: 91 80 852 0362
www.infy.com
mailto:E&R@infy.com

# Document Revision History

| Version | Date | Author(s) | Reviewer(s) | Comments |
|---|---|---|---|---|
| 1.0 | Jun-2010 | Anooja Mary Jacob | Bhoolakshmi M<br>Vijay Kumar Dani | FP Restructuring – 2010 |

# Contents

## Context

This document contains Practice Questions to be completed as part of the practice session for the subject Programming Practices.

## Guidelines

- The practice guide has been designed to give hands on experience to map the concepts learnt in the theory session with practical Practice Questions and provide **more hands on practice**
- The Practice Questions are organized based on topics
- These Practice Questions contain coding exercises, debugging exercises, coding standards exercises and many code snippets for analyzing and predicting the output
- Solving these exercises methodically would provide more practice and hence confidence to the learner to attempt the module and comprehensive exams
- There are three categories of Practice Questions –
  - √"Must to know"- The concepts in these Exercises are fundamental and must be practiced and known by the trainee
  -  "Need to know"- The concepts in these Exercises need to be learnt by the trainees and practiced
  -  "Good to know"- These concepts are some additional concepts which can be practiced by the trainees after completing the essential Assignments and exercises in the lab guide and threaded Exercises respectively.

# Must to know

## √Practice Question 1: Coding Standards and Best Practices

**Identify and correct the following code snippet for coding standards**

```
/*****************************************************************
* Filename     : CodingStandards.c
* Author       : Education & Research Dept, Infosys Technologies
*                Limited
* Date         : 16-May-2008
Description    : Helps to understand the coding standards for a program
*****************************************************************/

/* Include files */
#include<stdio.h>

/*****************************************************************
```

```
* Function       : main()
* Description    : main function of the program explains use of type
casting
Input Parameters:
int argc - Number of command line arguments
char **argv  The command line arguments passed
Returns: 0 on success to the operating system
*************************************************************************/

int main(int argc,char** argv){

/*Variable declaration */
int iValue = 10;
float iNumber = 2.0f;
double dEmpsalary = 10000;

/* Displaying the values */
printf("Integer  = %d\n",iValue);
printf("Float  = %0.2f\n",iNumber);
printf("Employee salary  = %0.2lf\n",dEmpsalary);

}
/*************************************************************************
* End of CodingStandards.c
*************************************************************************/
```

**Estimated Time: 10 minutes**

## √Practice Question 2: Use of while Loop

Find the output of the following programs by dry run through trace table and then compare the result by executing the program

(1)

```
#include<stdio.h>
int main(int argc,char **argv) {

  /*Variable declaration*/
  int iNumber = 5;
  while(1 == iNumber){
      iNumber = iNumber - 1;
      printf("%d ",iNumber);
      --iNumber;
  }
  getch();
  return 0;
 }
```

(2)

```
#include<stdio.h>
int main(int argc,char **argv) {
  int iNumber;
  while(iNumber = 0){

        iNumber++;
  }
    printf("%d ",iNumber);
  return 0;
 }
```

**Estimated Time: 15 minutes**

## √Practice Question 3: Use of do while Loop

Find the output of the following programs by dry run through trace table and then compare the result by executing the program

(1)

```
#include<stdio.h>
int main(int argc, char ** argv)
{
      int iIndex=1,iSum=0;
      do
      {
            iSum=iIndex*iIndex+iSum;
            printf("%d\n",iSum);
      }while(iIndex==1);

/* Return a success code to the Operating System */
      return 0;
}
```

(2)

```
#include<stdio.h>
int main(int argc, char ** argv)
{
      int iIndex=1,iSum=0;
      do
      {
            iSum=iIndex*iIndex+iSum;
            printf("%d\n",iSum);
      }while(iIndex=1);

/* Return a success code to the Operating System */
      return 0;
}
```

(3)

```
#include<stdio.h>
int main(int argc, char ** argv)
{
      int iIndex=1,iSum=0;

      do
      {
```

```
                    iSum=iIndex*iIndex+iSum;
                    printf("%d\n",iSum);
                }while(iIndex<4);

            /* Return a success code to the Operating System */
                return 0;
            }
```

**Estimated Time: 15 minutes**

## √Practice Question 4: 1D - Arrays

a) Dry run the following programs and also verify the result by executing the following programs

(1)

```
int main (int argc, char** argv){
      int aiEmployeeNumber[] = {15090, 15091, 15092};
      printf("%d  ", aiEmployeeNumber[1]);
      printf("\nenter the value for element\n");
      scanf("%d", &aiEmployeeNumber[1]);
       printf("%d  \n", aiEmployeeNumber[1]);
       return 0;
}
```

(2)

```
#include <stdio.h>

  int main(int argc,char**argv) {
  int aiArray1[2] = {1,2};
  int aiArray2[2] = {2,3};
  int iIndex;


  for(iIndex=0;iIndex<2;iIndex++){
    aiArray1[iIndex]=aiArray2[iIndex];
  }

  if(aiArray1==aiArray2){
    printf("They aiArray1re equal\n");
  }
  else{
    printf("They are not equal\n");
   }

  if(aiArray1==aiArray1){
    printf("Of course a is equal to a\n");
  }
  else {
    printf("No, a is not equal to a\n");
```

```
    }
    for(iIndex=0;iIndex<2;iIndex++){
      printf("aiArray1[%1d] = %3d\n", iIndex,
aiArray1[iIndex]);
 }
 return 0;
 }
```

**Estimated Time: 15 minutes**

## √ Practice Question 5 : A Complete Case Study

**Objective:** To implement a case study using the programming concepts learnt in this module

**Problem Statement:**

The Global Education Center library wants to keep track of the books in the library and update the number of copies as and when a new copy of an existing book arrives. As well as add a book's details as and when as new book arrives.

Write a menu driven program to implement the above scenario. The details are as given below:

**Information to be stored in the array of book structure:**

- Book code
- Book Title
- Book author
- No. of copies

**Note:**

- **The book code is given to a particular title of the book and not to each of the copies of the books. For example: For a book titled "Programming C", the book code would be 1001 and there could be 7 copies of the book**
- **When new books arrive , a new element needs to be added in the array of structures**
- **When copies arrive for the existing book, the number of copies are updated**

This program has three functionalities:

**Add Books**

- The Book Codes have to be auto generated starting from 1000. Every new book must have the Book code of the last unique book entered incremented by one
- Accept the Book Title

    - The Book Title should not exceed 15 characters and it should contain minimum 3 characters. The Book Title can contain only alphabets (upper case or lower case), blank space, hyphen (-) and ampersand (&). The first character in the Book Title should be an upper case alphabet
    - When an invalid Book Title is entered, display an error message and ask the user to reenter the Book Title
- Accept the Book Author
    - The Book Author should not exceed 15 characters and it should contain minimum 3 characters. The Book Author can contain only alphabets (upper case or lower case), blank space, hyphen (-) and ampersand (&). The first character in the Book Author should be an upper case alphabet
    - When an invalid Book Author is entered, display an error message and ask the user to reenter the Book Author
- Accept the Number of copies of the book
    - It should be greater than zero
    - If an invalid number of copy is entered, display an appropriate error message and ask the user to reenter the number of copies
- The Book Code, Book Title, Book Author and the No. of Copies should be written into the array of structures for book
- When the user finishes adding the new book, the control should return to the main menu after displaying a message "Book is Successfully Added to the library"

**Update Number of copies**

- Accept the Book Code
    - The accepted Book Code must exist in the array of structures, book
    - When an invalid Book Code is entered, display an error message and ask the user to reenter the Book Code
    - When a valid Book Code is entered, display the Book Author, Book Title and the existing copies of the books
- Accept the new Number of copies
    - It should be greater than zero
    - If an invalid Number of copy is entered, display an appropriate error message and ask the user to reenter the number of copies
- The number of copies entered is added to the existing number of copies and updated in the corresponding index in the array of structures for book
- When the user finishes updating the Number of copies entered, the control should return to the main menu after displaying a message "Copies are Successfully Updated"

**View all Books**

- Display all the book details such as Book Code, Book Title, Book Author and Number of copies in tabular format

**Estimated time: 60 minutes**

**Summary of this assignment:**
In this exercise, you have learnt:

- Implementing a case study using the programming concepts learnt in this module

## Need to know

### Practice Question 6: Variables, constants and data types

Compile and correct the code to have 0 warnings

```
/*************************************************************
* Filename    : Operators1.c
* Author      : Education & Research Dept, Infosys Technologies
*               Limited
* Date        : 04-Apr-2008
* Description : To understand the use of typecasting
*************************************************************/

/* Include files */
#include<stdio.h>
/*************************************************************
* Function      : main()
* Description   : main fuction to understand the use of
    *                  typecasting
* Input Parameters:
*     int argc - Number of command line arguments
*     char **argv  The command line arguments passed
* Returns: 0 on success to the operating system
*************************************************************/
int main(int argc,char **argv) {
 int iNumber = 10;
 double dValue=2.5;
 float fResult;
 fResult = iNumber + dValue;
 printf("Result = %f", fResult);
}
/*************************************************************
* End of Operators1.c
*************************************************************/
```

**Estimated Time: 10 minutes**

## Practice Question 7: Use of for Loop

a) Write a program to print 1 to 50 which are divisible by 4

b) Find the output of the following programs by dry run through trace table and then compare the result by executing the program

(1)

```
#include<stdio.h>
#define NUMBER 200

int main(int argc,char **argv) {

  /*Variable declaration*/
  int iCount1,iCount2,iFlag=0;

  printf("Prime numbers from 1 to 200 are: \n\n");
  /* loop from 1 to 200 */
  for(iCount1 = 1; iCount1 <= NUMBER; iCount1++){
         iFlag=0;
         /* loop to check whether number is prime or not */
       for(iCount2 = 2; iCount2 < iCount1; iCount2++){
              if(0 == iCount1 % iCount2){
                     iFlag=1;
                      break;
              }
         }
         /* check the iFlag value if it 1 the number is not prime*/
         if(0 == iFlag ){
                       printf("%d ", iCount1);
         }
  }
      return 0;
 }
```

(2)

```
#include<stdio.h>
int main(int argc,char **argv) {

  /*Variable declaration*/
  int iNumber,iIndex;

  printf("Enter the number upto which you want to display the
table\n\n");
  scanf("%d",&iNumber);
  printf("----------------------\n");
  printf("\nCharacter\tASCII value\n");
  printf("----------------------\n");

  /* loop to display characters and its equivalent ASCII values*/
  for(iIndex = 0; iIndex <= iNumber;iIndex++){
```

```
            printf("\n%c\t\t%d",iIndex,iIndex);
        }
            getch();
          return 0;
       }
```

**Estimated Time: 30 minutes**

## Practice Question 8: Use of while Loop

Find the output of the following programs by dry run through trace table and then compare the result by executing the program

(1)

```
#include<stdio.h>
int main(int argc,char **argv) {

  int iIndex=1;
  while(iIndex<6);
  {
      iIndex++;
    printf("%d ",iIndex);
  }
  getch();
  return 0;
 }
```

(2)

```
#include<stdio.h>
int main(int argc,char **argv) {
  int iIndex=1;
  while(iIndex<50)
  {
      iIndex++;
      if(iIndex%5==0)
            printf("%d ",iIndex);
  }
  getch();
  return 0;
 }
```

**Estimated Time: 15 minutes**

## Practice Question 9: Use of do while Loop

Write a program to print the message “Hello World” on screen till the user choice is ‘N’ or ‘n’ to stop the program

**Estimated Time: 15 minutes**

## Practice Question 10: 1D - Arrays

a) Dry run the following programs and also verify the result by executing the following programs

(1)

```
int main (int argc, char** argv){

   int aiEmployeeNumber[] = {15090, 15091, 15092, 15093};
   int *piPointer=NULL;
   piPointer = aiEmployeeNumber;
   printf("\n%d  ", aiEmployeeNumber[0]);
   printf("\n%d  ", piPointer[0]);
   printf("\n%d  ", *(piPointer + 0));
   printf("\n%d  ", *piPointer);
   printf("\n\n%d  ", *(piPointer + 2));
  return 0;
}
```

(2)

```
#include<stdio.h>
#define N 10

int main(int argc,char **argv){
int iNumber[N],iSize,iCount1,iCount2,iTemp;
printf("\n how many no's in array");
scanf("%d",&iSize);
printf("\n enter %d elements in array",iSize);
for(iCount1=0;iCount1<iSize;iCount1++){
  scanf("%d",&iNumber[iCount1]);
}
for(iCount1=0;iCount1<iSize;iCount1++){
      for(iCount2=iCount1+1;iCount2<iSize;iCount2++){
      if(iNumber[iCount1]< iNumber[iCount2]){
          iTemp=iNumber[iCount1];
      iNumber[iCount1]=iNumber[iCount2];
      iNumber[iCount2]=iTemp;
      }
}
printf("%4d",iNumber[iCount1]);
}
getch();
}
```

**Estimated Time: 15 minutes**

## Practice Question 11: 2D - Arrays

a) Understand the following code and fill the logic for the one in red in font

```
int main (int argc, char** argv){

/* Declare a 2D array aiEmployeeInfo of size 3 x 5,1st row for
holding Employee Id, 2nd row is for Department code and 3rd row
is for salary. Initialize the first row with
1001,1002,1003,1004,1005 */

     <Declare the array here>
     int iEmpId,iFound =0,iCount;
     char cChoice;

 /* Enter the employee id */
     printf("Enter the Employee id\n");
     scanf("%d", &iEmpId);

     do{
         for(iCount=0; iCount<5;iCount++){
 /* check whether read Employee Id is present in the array or not */
          if(< condition >){
               printf("Employee Id is present\n");
               iFound = 1;
               break;
          }
          }
        /* Getting department code and salary for the found employee id
        */
   if(1 == iFound){
      printf("Enter the Department code in the range 101-
105\n");
       scanf("%d",&aiEmployeeInfo[1][iCount]);
       printf("Enter the Salary\n");
       scanf("%d",&aiEmployeeInfo[2][iCount]);
      }
    else{
       /* Display error message and terminate the program */
       printf("Employee Id not Found\n");
       return 0;
       }
    /* Read the choice from user to continue or not */
       printf("Do you want to continue(Y/y)");
       fflush(stdin);
       scanf("%c",&cChoice);
    }while(cChoice == 'Y' || cChoice == 'y');

  /* Return a code back to OS!*/
   return 0;
}
```

**Estimated Time:15 minutes**

## Practice Question 12: Strings

Accept a string and check whether it contains only alphabets (a-z or A-Z)

**Estimated Time:15 minutes**

# Good to know

## Practice Question 13 : Conditional constructs

a) Write a program to perform arithmetic operations on the given numbers. i.e. Addition, Subtraction, Multiplication and Division operations on the given numbers. Accept the choice of operation from the user. Convert this into a program

b) What is the output of the following code snippet?

```
int main(int argc,char **argv) {
  if(-2){
    printf("-2 means true");
    }
  if(0){
    printf("0 means true");
    }
  if(1){
    printf("1 means true");
 }
}
```

**Estimated Time:20 minutes**

## Practice Question 14: Use of while Loop

a) Write a program to accept an integer and extract every digit in the integer and print them in words in the reversed order.

b) Find the output of the following programs by dry run through trace table and then compare the result by executing the program

(1)

```
#include<stdio.h>
int main(int argc,char **argv) {

  /*Variable declaration*/
  float fNumber = 2.5f;;
```

```
        while(fNumber == 2.5f){
            printf("%.2f ",fNumber);
            fNumber++;
        }
        getch();
        return 0;
       }
```

(2)

```
      #include<stdio.h>
      int main(int argc,char **argv) {
            int iFirstNumber = 4,iSecondNumber = 0,iValue;

            while(iFirstNumber >= 0){

                  if(iFirstNumber == iSecondNumber){
                        break;
                  }
                  else{
                       printf("%d %d\n",iFirstNumber,iSecondNumber);
                  }
                  iFirstNumber--;
                  iSecondNumber++;
            }
            getch();
            return 0;
       }
```

(3)

```
      #include<stdio.h>
      int main(int argc,char **argv) {

        /*Variable declaration*/
        float fOverTimePay;
        int iHour,iIndex=1;

        /* Over time pay calculation for 3 employees */
        while(iIndex <= 3){

             printf("Enter the no. of hours worked\n");
             scanf("%d",&iHour);
            /* Over time pay calculation */
             if(iHour >= 40){
                  fOverTimePay = (iHour - 40) * 12;
                  printf("No. of Hours worked = %d\n \
                   OvertimePay = Rs. %f",iHour,fOverTimePay);
             }
             else{
                   fOverTimePay = 0;
                 printf("\nNo. of hours worked (%d) is less \
                 than 40hrs.\nHence no overtime pay ",iHour);
             }
             iIndex++;
        }
            return 0;
      }
```

(4)

```
#include<stdio.h>
int main(int argc,char **argv) {

  /*Variable declaration*/

  int iIndex=1,iFirstNumber,iSecondNumber,iThirdNumber;
  printf("ARMSTRONG NUMBERS BETWEEN 1 & 500 ARE\n");

  while(iIndex <= 500){
        /* Extract last digit */

        iFirstNumber = iIndex % 10;
        iSecondNumber = iIndex %100;
        /*Extract second digit */
        iSecondNumber = (iSecondNumber - iFirstNumber)/10;
        /* Extract first digit */
        iThirdNumber = iIndex / 100;
       if((iFirstNumber*iFirstNumber*iFirstNumber) +
      (iSecondNumber*iSecondNumber*iSecondNumber) +
      (iThirdNumber*iThirdNumber*iThirdNumber)== iIndex){
            printf("%d ",iIndex);
       }
        iIndex++;

  }
    return 0;
}
```

**Estimated Time:30 minutes**

## Practice Question 15: 1D - Arrays

a) Write a program to find the number of occurrences of a given number in an array of elements

b) Create an array that stores the marks scored by students in a subject. You can assume the size of the student array to be 10. The range of the marks must be 0 to 25. Write a program to accept the marks of the students and count the number of students who have scored each marks in the range, i.e you need to count how many students have scored 0, how many students scored 1, how many students scored 2 ... how many students scored 25

c) Create two integer arrays. merge the elements of both the arrays into the third array as given in the example below:

Ex: Array1 contains { 0,1,2,3,4 }
Array2 contains { 5,6,7,8,9 }
Resultant array contains { 0,5,1,6,2,7,3,8,4,9}

**Estimated Time:45 minutes**

## Practice Question 16: 2D – Arrays & Functions

a) Create an array that stores the names of 5 students. Create a 2D array that stores the marks of 5 students in 5 different subjects. Write a function to find the total marks of the students and also to find which student has scored the highest total.

Invoke the function from the main method as and when it is required.

b) Create a 2D array that stores the sales id and sales in rupees made by the salespersons during the month. (Assume that the sales id and sales are whole numbers).

Each salesperson gets a salary of Rs.1000 plus 10% commission (commission on the sales made, Ex: if the sales is 3000 then commission is 10%of 3000) only if the sales made during the month crosses Rs.3000. Otherwise his salary is zero. Write a function to calculate the salary of the sales persons and populate it into another array.

Write a function to print in a neat format how many sales people earned salaries in each of the following ranges:

Rs.1000 to Rs.2000
Rs.2001 to Rs.3000
Rs.3001 to Rs.4000
Rs.4001 to Rs.5000

Invoke the functions from the main method as and when it is required.

**Estimated Time: 45minutes**