Punjab University College of Information Technology (PUCIT), University of the Punjab

Programming Fundamentals – Spring 2012

Final Exam

Date: Monday, July 02, 2012

Time Allowed: **2 Hours** Total Marks: **80**

Roll No. _____

Student's Name: _____

Student's Signature: _____

Answer Book No:
Invigilator's Signature:

Tick Your Section

BSSE	*MOR*
	AFT

BSIT	*MOR*
	AFT

BSCS	*MOR-A*	*MOR-B*
	AFT-A	*AFT-B*

INSTRUCTIONS:

1. You are NOT allowed to ask any questions during the exam. If there is some mistake in the question paper, the benefit goes to the student.
2. Use of calculator is NOT allowed.
3. Use only blue or black ink to solve the exam. Even the lead pencil is also not allowed.
4. Suggested time is mentioned for each question. Manage your time accordingly.
5. Write your answers and working within the given space for each question.
6. You may use any blank/unused space on the exam booklet for rough work. No extra sheet will be provided for rough work.
7. Make sure that there are **8 double-side sheets** (16 pages) in this exam, including this title sheet.
8. When you are told to open this booklet, first of all write your ROLL NUMBER on top of each sheet.
9. Do not forget to keep praying before and during the exam. And remain honest! ☺

Question #	Q1	Q2	Q3	Q4	Q5	Q6	TOTAL
Total Marks	15	10	10	15	15	15	80
Marks Obtained							

DO NOT OPEN THIS BOOKLET
UNTIL YOU ARE TOLD TO DO SO!

☺ **Good Luck!** ☺

Examiner's Signature:

This page is left blank intentionally
(You may use it for your rough work)

Roll No. _____

QUESTION # 1 **15 Marks** **Suggested Time: 15 minutes**

Select (encircle) the best option for each of the following. **Each cutting/overwriting will give you -0.5**, in this question.

1. What is the result of the expression **`(5 | 3)`**?
   A) 1 (means true)
   B) 0 (means false)
   C) 7
   D) 8

2. What is the result of the expression **`('A'&'a')`**? The ASCII code of **`'A'`** is 65, and that of **`'a'`** is 97
   A) 65 (means ASCII of 'A')
   B) 97 (means ASCII of 'a')
   C) 32
   D) None of the above

3. What is the result of the expression **`~5`** ?
   A) 5
   B) -5
   C) 0
   D) None of the above

4. What is the result of the expression **`5 >> 2`** ?
   A) 5
   B) 2
   C) 1
   D) 0

5. What is the result of the expression **`5 << 2`** ?
   A) 5
   B) 2
   C) 20
   D) 10

<u>Question # 6, 7, and 8</u> use the following enumeration:

```
enum Days {FRI, THU, WED, TUE, MON, SUN, SAT};
```

6.
```
int a=5;
switch (a)
{
   case 5-1:  cout << SUN;
   case 5:    cout << MON;
   case 5+1:  cout << TUE;
   default:   cout << "WED";
}
```
What is the output of the above snippet of C++ code ?
   A) MON
   B) MONTUEWED
   C) 432
   D) None of the above

7. `if (SUN > MON) cout << SUN; else cout << MON;`
   What is the output of the above snippet of C++ code ?
   A) SUN
   B) MON
   C) 4
   D) 5

8. `if (SUN==2) cout << TUE; else cout << WED;`
   What is the output of the above snippet of C++ code ?
   A) 3
   B) 2
   C) TUE
   D) None of the above

9. Which of the following are valid syntax in C++: **[You may select one or more as correct]**
   A) `enum {a, b, c};`
   B) `enum {d=2, e=1, f=0};`
   C) `enum {x, y=5, z};`
   D) `enum Day = {SUN};`

10. `enum month {J, F, Mr, Ap, M, Ju, Jy, A, S, O, N, D} a, b, c;`
    `int x=7;`
    Which of the following operations are valid: **[You may select one or more as correct]**
    A) `c = a + b;`
    B) `b = a;`
    C) `c = x;`
    D) `x = b;`

Question # 11 and 12 relate to the following code:

```
int * ip, * jp;
int arr [] = {0, 1, 2};
ip = arr;
jp = arr+2;
```

11. What is the output of `cout << ip - jp;` after the above snippet of C++ code ?
    A) 2
    B) 8
    C) 4
    D) None of the above

12. What is the output of `cout << *ip - *jp;` after the above snippet of C++ code ?
    A) 1
    B) 2
    C) -2
    D) None of the above

<u>Question # 13 and 14</u> relate to the following code:

```
struct S {int * ip, *jp; int arr[3];};
S *a, b;
a = &b;
b.ip = b.arr;
b.jp = b.arr+2;
b.arr[0]=0; b.arr[1]=1; b.arr[2]=2;
```

13. What is the output of **`cout << a->ip - a->jp;`** after the above snippet of C++ code ?
    A) 2
    B) 8
    C) 4
    D) None of the above

14. What is the output of **`cout << *b.ip - *b.jp;`** after the above snippet of C++ code ?
    A) 1
    B) 2
    C) -2
    D) None of the above

15. What is the output of **`cout << sizeof(a);`** after the above snippet of C++ code ?
    A) 20
    B) 24
    C) 12
    D) None of the above

This space is left blank intentionally
(You may use it for your rough work)

QUESTION # 2 **5x2 = 10 Marks** **Suggested Time: 15 minutes**

Write the output for the following five code segments:

a)
```
int array[5] = {4,7,2};
int *ptr = array;
cout << *ptr+3 << " ";
cout << *(ptr+3) << " ";
cout << *(array+2) << " ";
cout << *(array+*ptr) << " ";
```

Output:

b)
```
int i, j;
int *p1, *p2;
p1 = &i;
p2 = &j;
i = 50;
j = 60;
cout << *p1 << " " << *p2 << endl;
p1 = p2;
i = *p1;
*p1 = *p2 + 1;
cout << i << " " << j;
```

Output:

c)
```
struct data
{
      int * p;
};
int main()
{
      data dt;
      dt.p = new int[3];
      for (int j = 0; j < 3 ; j = j+1)
      {
            dt.p[j] = j+1;
      }
      cout << *((dt.p)++) << " ";
      cout << *dt.p << " ";
      cout << *((dt.p)++) << " ";
      cout << *dt.p << " ";
      return 0;
}
```

Output:

d)
```
enum months {JAN, FEB, MAR, APR, MAY, JUN,
JUL, AUG, SEP, OCT, NOV, DEC};

months cur_month, next_month, prev_month;

cur_month = JUL;
prev_month = static_cast<months> (cur_month - 1);
next_month = static_cast<months> (prev_month + 2);

cout << prev_month << " ";
cout << cur_month << " ";
cout << next_month << " ";
```

Output:

_____

_____

e)
```
int i, j, k;
i = 4;
j = 3;

k = (i ^ j) << 1;
cout << k << " " << i << endl;

j = (k & i) >> 2;
cout << j << " " << k << endl;
```

Output:

_____

_____

This space is left blank intentionally
(You may use it for your rough work)

QUESTION # 3 **5x2 = 10 Marks** **Suggested Time: 15 minutes**

There is **exactly one syntax error** in each of the following five code segments. You are required to identify the line containing error and re-write that line correctly. [10]

a)	1	`int* fn_add (int *p1, int p2)`
	2	`{`
	3	`int s = *p1 + p2;`
	4	`return s;`
	5	`}`

Line # *Corrected code*

_____ _____

b)	1	`int arr[5] = {1,2,3,4,5};`
	2	`for(int i = 0; i < 5; i=i+1)`
	3	`{`
	4	`cout << *arr++ << endl ;`
	5	`}`

Line # *Corrected code*

_____ _____

c)	1	`struct point`
	2	`{`
	3	`int x = 5;`
	4	`};`
	5	`point p;`
	6	`cout << p.x << endl;`

Line # *Corrected code*

_____ _____

d)	1	`struct matchBox`
	2	`{`
	3	`int matchSticks;`
	4	`};`
	5	`matchBox * mB;`
	6	`mB = new matchBox;`
	7	`*mB.matchSticks = 50;`

Line # *Corrected code*

_____ _____

e)	1	`fstream file;`
	2	`file.open("myfile.dat", ios::in | ios::binary);`
	3	`int s;`
	4	`file.read(&s, sizeof(s));`
	5	`cout << s << endl;`

Line # *Corrected code*

_____ _____

This space is left blank intentionally
(You may use it for your rough work)

QUESTION # 4 **8+7 = 15 Marks** **Suggested Time: 15+10 = 25 minutes**

***TOPIC:** C-strings, Character case conversion*

A) Write a **C++ function** which takes the **first name** of the employee (a c-string containing at most 15 characters), and his/her **department** (CS, IT, or SE) of PUCIT from user. Then, the function should compose the email ID of the employee like this:

`firstName@dept.pucit.edu.pk`

The composed email ID of the employee must be stored in a c-string and then displayed on screen. Two sample runs of the program are shown below:

Sample Run 1:

```
Enter first name: Ali
Enter department: CS

Your email ID is: ali@cs.pucit.edu.pk
```

Sample Run 2:

```
Enter first name: GaDdis
Enter department: It

Your email ID is: gaddis@it.pucit.edu.pk
```

Note that the user can enter his/her **name** and **department name** in CAPITAL, small, or mIxEd case. But, all the letters of the email ID MUST be in **lower case**. [8]

```
void generateEmailID ()
{
    // complete the function definition here
```

You may continue the answer of this question on next page, if needed.

You may use empty space of this page for rough work, also.

***TOPIC:* C-strings, String/numeric conversion functions**

B) Write a C++ function **`findMax`** that receives an **array of c-strings** and the **size of array** as parameters. Each c-string contains a positive integer (stored as characters). The function will determine and return the largest of all of these numbers. [7]

```
int findMax (char * nums[], int size)
{
    // complete the function definition here
```

Roll No. _____

QUESTION # 5 **15 Marks** **Suggested Time: 20 minutes**

***TOPIC:* Pointers, Dynamic memory allocation**

Write a C++ function which takes an array of **`double`**s and its size as arguments. Firstly, the function should determine the number of positive values present in the array. After that the function should **dynamically allocate** an array of **`double`**s to store those many values. Then, the function will copy **all the positive values** from the original array into the newly allocated array. At the end, the function should return a pointer to the dynamically allocated array containing all the positive values. The size of this new array will be returned through a reference parameter (see the 3rd argument in the function prototype given below).

The prototype of your function should be:

```
double* extractPositives (double* original, int origSize, int& newSize);
```

Note: *Indent your code properly. Use meaningful variable names.*

```
double* extractPositives (double* original, int origSize, int& newSize)
{
    // complete the function definition here
```

You may continue the answer of this question on next page, if needed.

You may use empty space of this page for rough work, also.

Roll No. _____

QUESTION # 6 **15 Marks** **Suggested Time: 25 minutes**

***TOPIC:* Pointer to structures, File I/O (Text, Binary)**

You are given a **text file** (**`accounts.txt`**), which contains information about various accounts of a bank. The first line of this text file contains an integer indicating the number of account records present in the file. Each account's information is stored on two lines in the input file. The first line contains the **Account ID** (an integer) and the **Name of Account Holder** (a c-string containing at most 50 characters). The second line contains the **Current Balance** (a double value) of the account. A sample input file is shown below:

```
3
760 Ali Baig
40250.60
136 Saleem Arshad
35624
545 Javed Iqbal
41804.5
```

You are required to write a C++ program which reads the accounts' information from the input **text file** (**`accounts.txt`**) and stores all the information in a **binary file** (**`accounts.dat`**). You are required to use the following struct **`Account`** to store all information about a particular account:

```
struct Account {
      int accountID;  char name [51]; double balance;
};
```

The flow of your program would be:

After opening the input *text file* (**`accounts.txt`**) your program should read number of Accounts and then dynamically allocate an array of **`Account`**s. It should then read the records of accounts from the text file and store them in this array. When all the records have been read, your program should close the input file. After this, your program will create/open a *binary file* (**`accounts.dat`**), and store all the information read from the input file (i.e. the number of records and those many records of **`Account`** type) into it. Finally, this binary file will be closed by your program, and all the dynamically allocated memory will be deallocated.

```
#include <iostream>
#include <fstream>
using namespace std;
struct Account {
     int accountID;    char name [51];     double balance;
};
int main ()
{      // complete the function definition here
```

Continue the answer of this question on next page.

You may use empty space of this page for rough work, also.