



Khalifa University of Science, Technology and Research
Faculty of Engineering
CMPE111 – Principles of Computer Programming
Final Exam
Spring 2010 - 2011

Monday, 6th of June, 2011

DO NOT WRITE YOUR NAME OR ID ON THIS PAGE

Duration: 3 Hours.

Question 1 42 marks	
Question 2 12 marks	
Question 3 7 marks	
Question 4 10 marks	
Question 5 17 marks	
Question 6 12 marks	
Total Grade 100	

Instructions:

- ✓ Check to insure that you have a complete copy of the examination. There are 6 Questions, and a total of 10 pages
- ✓ This examination is closed book **and** closed notes.
- ✓ Questions are NOT allowed during the exam.
- ✓ The use of electronic calculators or other equipment is not allowed during the examination.
- ✓ Scratch papers are NOT allowed. You should work your Questions directly on the Question statement sheet in the appropriate space only. You may use the backside of the exam sheets as appropriate.
- ✓ It is recommended that you read the whole exam before attempting any Question. Also, please don't spend more than the proposed time for each Question.
- ✓ Include all steps as appropriate for maximum credit. Be clear, brief, and specific in your answers.

Good Luck!

Question 1: (42 marks) *proposed time 60 min*

Find the output of the following programs?

1.

```
#include <iostream>
using namespace std;
void Update (int a, int &b)
{
    a=++b+a;
}
int main ()
{
    int a=6, b=7;
    Update(b,a);
    cout<<a<<" " <<b<<endl;
    return 0;
}
```

Solution
2.

```
#include <iostream>
using namespace std;
int main()
{int x, y;
  x=20;
  y=4;
  cout << (x/y++)%3<<endl;
  return 0;}
```

Solution
3.

```
#include <iostream>
using namespace std;
int main()
{
    int grade[7]= {10, 8, 5, 7, 6, 3, 9};
    int Sum=0;
    for (int i=0; i<grade[5]; i++)
        Sum=Sum + grade[i];
    cout<< Sum<<endl;
    return 0;
}
```

Solution
4.

```
#include <iostream>
using namespace std;
int main()
{int a [ 3][3 ] = {{40, 60, 50},{20, 40, 60}, {30, 20, 70}};
  int sum=0;
  int j=1;
  for(int i=2; i>=0; i--)
      sum= sum + a[i][j];
  cout<<sum<<endl;
  return 0; }
```

Solution

5. Solution
- ```
#include <iostream>
using namespace std;
int main()
{
 int x=5, y=3, z=4;
 switch(z%x*y)
 {
 case 10: cout<<"+++++"<<endl;break;
 case 12: cout<<"#####"<<endl; break;
 case 13: cout<<"$$$$$"<<endl;
 case 14: cout<<"*****"<<endl; break;
 }
 return 0;
}
```
6. Solution
- ```
#include <iostream>
using namespace std;
int main()
{
    int i;
    for (i=15; i>=1; i=i-3)
    {
        if(i % 9 == 0)
            break;
        cout<<i<<endl;
    }
    return 0;
}
```
7. Solution
- ```
#include <iostream>
using namespace std;
int main ()
{
 int x = 5, y = 4;
 int *p1;
 p1 = &x;
 *p1 = y++;
 cout << "firstvalue is " << x << endl;
 cout << "secondvalue is " << y << endl;
 return 0;
}
```

8. `#include <iostream>`  
`using namespace std;`  
`int function1(int a, int b=5)`  
`{`  
`return a%b;`  
`}`  
`int main ()`  
`{`  
`cout<<function1(4, 7)<<endl;`  
`cout<<function1(5)<<endl;`  
`return 0;`  
`}` Solution
9. `#include <iostream>`  
`#include <string>`  
`using namespace std;`  
`int main()`  
`{ string x="HLEOL";`  
`for (int i=1; x[i]!=NULL; i+=2)`  
`x[i]='-';`  
`cout<<"The new string is: "<<x<<endl;`  
`return 0;`  
`}` Solution
10. `#include <iostream>`  
`using namespace std;`  
`int main()`  
`{int sum=0;`  
`int a[6]= {10,20,5,14,30,49};`  
`for(int i=5;i>=1;i--)`  
`{if(a[i]%7==0)`  
`continue;`  
`sum=sum + a[i];`  
`}`  
`cout<<sum<<endl;`  
`return 0; }` Solution
11. `#include <iostream>`  
`using namespace std;`  
`int main()`  
`{`  
`char x [40]="Principles-of-Computer-Programming";`  
`int i = 14;`  
`do`  
`{`  
`cout<<x[i];`  
`}while (x[i++] != 'r');`  
`cout<<endl;`  
`return 0;}` Solution

12. `#include <iostream>`  
`using namespace std;`  
`int function1 (int n);`  
`int main()`  
`{`  
`int a, b=1;`  
`a = function1(b);`  
`cout<< a;`  
`return 0;`  
`}`  
`int function1 (int n)`  
`{`  
`if (n==5)`  
`return n;`  
`return n*function1(n+1);`  
`}`
13. `#include <iostream>`  
`using namespace std;`  
`int main()`  
`{`  
`int i=4;`  
`int j=10;`  
`if (i+j<15)`  
`{`  
`int i=10;`  
`cout<<j+i<<" ";`  
`j--;`  
`}`  
`cout<<j-i <<endl;`  
`return 0;`  
`}`
14. `#include <iostream>`  
`using namespace std;`  
`int main()`  
`{`  
`for (int i=9; i>=1; i=i-2 )`  
`{`  
`for (int j=0; j<=i; j=j+3)`  
`cout<<"*";`  
`cout<<endl;`  
`}`  
`return 0;`  
`}`
- Solution
- Solution
- Solution

**Question 2: (12 marks)** \_\_\_\_\_ *proposed time 20 min*

Diabetes is a chronic disease that has two major types: type 1 and type 2. A hospital wants to classify patients according to diabetes type so that all patients with same diabetes type are classified in the same file. The data of the patients are all saved in "Patient.dat" file where rows represent patients and columns represent patient's ID, Name, and type of diabetes which is either Type1 or Type2.

Patient.dat

|      |         |       |
|------|---------|-------|
| 1000 | Mohamed | Type1 |
| 1010 | Sara    | Type1 |
| 2001 | Omar    | Type2 |
| 3000 | Hamdan  | Type1 |
| 4030 | Ahmad   | Type2 |
| ...  |         |       |
| .    |         |       |

- a- Write the C++ code that reads the patient's data from the "Patient.dat" file until reaching the end of file. Note that the number of patients is unknown. After reading, you have to classify the patients according to their diabetes type where all patients with diabetes type1 are saved into "Type1.dat" file and all patients with diabetes type2 are saved in another file called "Type2.dat". Example of the output files. [9 marks]

| Type1.dat    | Type2.dat  |
|--------------|------------|
| 1000 Mohamed | 2001 Omar  |
| 1010 Sara    | 4030 Ahmad |
| 3000 Hamdan  | ...        |
| ...          | .          |
| .            |            |

- b- At the end of your program print on the screen the number of patients with diabetes type1 and number of patients with type2. [3 marks]

**Question 3: (7 marks)** *proposed time 10 min*

Consider the prototype of a function called **sum** where **A** is an array of integer numbers, **s** is the size of the array **A** and **n** is an integer number:

**int sum (int A[], int s, int n);**

Write the definition of the function **sum** to add the array elements located in the positions (indexes) with the same parity as the number **n** (the third function parameter).

Example: Consider the following array: `int A[7]={12,43,76,82,243,56,78};`

1) If you call the function as: **sum** (A,7,34); since the last argument 34 is an even number this function call will compute the sum of the elements in the positions (indexes) 0,2,4,6 which are 12, 76, 243 and 78. The function will return 409 (=12+76+243+78).

2) If you call the function as: **sum** (A,7,11); since the last argument 11 is an odd number this function call will compute the sum of the elements in the positions (indexes) 1,3,5 which are 43, 82 and 56. The function will return 181 (=43+82+56).

**Question 4: (10 marks)** \_\_\_\_\_ *proposed time 15 min*

A string is an array of characters that is terminated by the NULL character ‘\0’.

- a) Write a function **void Reverse (char S[])**, that reverses a string S. The function must receive a string S and print its reverse. For example: [5 marks]

| Before reverse string S | After reverse string S |
|-------------------------|------------------------|
| CMPE111                 | 111EPMC                |

Note: If string S is printed on the screen it will appear as: 111EPMC

- b) Also, you have to write a function **int VowelCount(char s[])** that counts the number of vowels in the string s. The function must return the number of vowels in the string s. [5 marks]

**Question 5: (17 marks)** *proposed time 30 min*

Write the C++ code that allows you to store the Patient's data in a one dimensional array of struct.

a) For this purpose, define the struct **Patienttype** that contains:[4 marks]

- i. **ID**: integer
- ii. **Name**: string
- iii. **Weight**: double
- iv. **Height**: double
- v. **BMI**:double (It is the body mass index =  $\text{Weight}/\text{Height}^2$  )
- vi. **Overweight**: Boolean

**Example:**

| ID   | Name   | Weight (Kg) | Height (m) | BMI               | Overweight |
|------|--------|-------------|------------|-------------------|------------|
| 1000 | Ali    | 100         | 1.8        | $100/1.8^2=30.86$ | True       |
| 2000 | Ahmad  | 79          | 1.79       | $79/1.79^2=24.68$ | False      |
| 1500 | Rana   | 70          | 1.82       | $70/1.82^2=21.14$ | False      |
| 2500 | Hamdan | 110         | 1.7        | $110/1.7^2=38.06$ | True       |
| 3000 | Abed   | 89          | 1.9        | $89/1.9^2=24.65$  | False      |

- b) Write the C++ code that allows you to read the data of **10** patients **except** the **BMI and Overweight** that will be calculated later in part (c). The data must be saved into one dimensional array of Patient. [4 marks]
- c) Write the C++ code that calculates the BMI and overweight of each patient. The Body Mass Index (BMI) is =  $\text{Weight}/\text{Height}^2$ . If the BMI is greater than 25 then the patient is overweight and the overweight parameter must set to "true". [5 marks]
- d) Write the C++ code that prints all the overweight patients including ID, Name, weight, height, BMI. [4 marks]

Example of output:

| ID   | Name   | Weight (Kg) | Height (m) | BMI               |
|------|--------|-------------|------------|-------------------|
| 1000 | Ali    | 100         | 1.8        | $100/1.8^2=30.86$ |
| 2500 | Hamdan | 110         | 1.7        | $110/1.7^2=38.06$ |

**Question 6: (12 marks)** *proposed time 20 min*

Consider a two dimension integer array with same **odd** sizes. For example the array can be as one of the following: int **num**[n][n]; where n=3, 5, 7, or 9 etc.

1. Write the code (you are free to use functions or without functions) to display the sum of the elements in the middle row.

For example, if the array is int **num** [3][3]; and its elements are: [6 marks]

```
1 3 5
8 4 6
9 7 9
```

You have to compute the sum of the numbers in bold: 8+4+6 (=18) which is the sum of the elements in the second row of the array **num**. If the array is int **num** [5][5]; and its elements are:

```
1 2 3 4 5
0 2 5 4 7
3 2 1 5 6
9 8 7 6 5
8 6 5 4 3
```

You have to compute the sum of the numbers in bold: 3+2+1+5+6 (=17) which is the sum of the elements in the second row of the array **num**.

2. Write the code that can swap the elements in the middle row with all the elements in the last row of a given array of any size. For example the 5 \* 5 array of the previous example will be after swap: [6 marks]

```
1 2 3 4 5
0 2 5 4 7
8 6 5 4 3
9 8 7 6 5
3 2 1 5 6
```