## CPE 150 INTRODUCTION TO PROGRAMMING FIRST "MAKEUP" EXAM

## Department of Computer Engineering Yarmouk University August 3, 2017

This is a CLOSED BOOK exam. Textbooks, notes, laptops, calculators, personal digital assistants, cell phones, and Internet access are NOT allowed.

It is a 60 minute exam, with a total of 15 marks. There are 4 questions, and 6 pages (including this cover page). Please read each question carefully, and write your answers legibly in the space provided. You may do the questions in any order you wish, but please USE YOUR TIME WISELY. When you are finished, please hand in your even paper and sign out. Cood hudd

When you are finished, please hand in your exam paper and sign out. Good luck!

Name: \_\_\_\_\_

Student I.D.:

Instructor and Section:

**Q1.** (3 marks) Complete the following C++ program that reads two integer values from the user x and y. Then, round the value of x to the nearest multiple of y. Listings 1 and 2 show two sample runs of the intended program.

```
[Note: assume that x > y and x, y \neq 0]
```

Enter two positive integers: 25 6 Result: 25 is rounded to 30

> Listing 1: Sample Output 1 for O1:

Enter two positive integers: 53 7 Result: 53 is rounded to 56

Listing 2: Sample Output 2

```
Answer for Q1:
```

}

```
int main() {
    int x, y, result;
    cout << "Enter two positive integers: ";
    cin >> x >> y;
```

cout << "Result: " << x << " is rounded to " << result << endl; return 0;

**Q2.** (4 marks) Complete the following C++ program that reads an integer and returns the term in the following sequence corresponding to that number. The sequence is defined as follows:

$$f(x) = \begin{cases} 1 & x < 0 \\ x & x = 1 \text{ or } x = 2 \\ f(x-1) + f(x-2) \times f(x-3) & x > 2 \end{cases}$$

Based on that definition, the series goes as follows:  $1, 1, 2, 3, 5, 11, 26, 81, \dots$ , so if the user enters a 0 or negative integer the result would be 1, if the user enters 1 or 2 the result would be 1 or 2. If the user enters 6, then the result would be 26 = f(5) + f(4) \* f(3) and so on. Listings 3 and 4 show two sample runs of the intended program.

```
Enter number: 6
1 1 2 3 4 11 26
Result: 26
```

```
Enter number: 5
1 1 2 3 5 11
Result: 11
```

Listing 3: Sample Output 1

Listing 4: Sample Output 2

Your program should **not** use any recursive functions or arrays. Your program must use a loop to produce the result as well as printing the terms until the desired number.

Answer for Q2:

```
int main() {
    int x, result;
    cout << "Enter number: ";</pre>
    cin >> x;
    cout << "Result: " << result << endl;</pre>
    return 0;
```

Q3. (4 marks) Complete the following C++ program that reads two positive integers and prints out the prime numbers between them. A prime number is a number that is only divisable by 1 and itself. For example, if the user enters 0 and 10, then the program prints the numbers: 2 3 5 7, if the user enters 20 10, the the program prints the numbers: 11 13 17 19. If one of the numbers is negative, the program should output Wrong Input!. Listings 5 and 6 show two sample runs of the intended program.

";

Enter two numbers: 0 10 2 3 5 7

Listing 5: Sample Output 1

Enter two numbers: 20 10 11 13 17 19

Listing 6: Sample Output 2

## Answer for Q2:

```
int main() {
    int x, y;
    cout << "Enter two numbers:</pre>
    cin >> x >> y;
```

return 0;

Q4. (4 marks) Complete the following C++ program that reads a 6-digit integer from the user and determines whether the number is *palindrome*. A *palindrome* number is a number that can be read from left to right the same as from right to left. For example, if the user enters 123321 then the number is palindrome. If the user enters 123456, then the number is *not* palindrome. Listings 7 and 8 show two sample runs of the intended program.

| Ente | er | а  | 6-c | ligit | nu | ımbe | er:            | 123456     |   |
|------|----|----|-----|-------|----|------|----------------|------------|---|
| The  | nu | mb | er  | 12345 | 56 | is   | $\mathtt{not}$ | palindrome | ! |

Enter a 6-digit number: 123321 The number 123321 is palindrome!

Listing 7: Sample Output 1

Listing 8: Sample Output 2

Your program should **not** use any recursive functions or arrays.

## Answer for Q4:

```
int main() {
    int x;
    cout << "Enter a 6-digit number: ";
    cin >> x;
```

return 0;

| + Data Typesia TypeDescriptionarCharactersigned charUnsigned CharacterttUnsigned CharacterttSint integerortSame as short intsigned short intUnsigned short integerortSame as unsigned intsigned intUnsigned intuntUnsigned long intsigned intUnsigned long intsigned longSame as unsigned intndLong integersigned longSame as unsigned long intoatUnsigned long intoatSame as unsigned long intoatUnsigned long intoatSame as unsigned long intoatUnsigned long intoatSame as unsigned long intoatSame as unsigned long intoatUnsigned long intoatSame as unsigned long intoatSame as unsigned long intoatSame as unsigned long intoatSame as unsigned long int <th>Commonly Used Operators<br/>Assignment Operators<br/>Assignment Operators<br/>Assignment Operators<br/>= assignment<br/>= combined subtraction/assignment<br/>= Combined mulpication/assignment<br/>= Combined mulpication/assignment<br/>= Combined modulo/assignment<br/>= Subtraction<br/>= Mulpication<br/>= Mulpication<br/>= Mulpication<br/>= Less than or equal to<br/>= Less than or equal to<br/>= Less than or equal to<br/>= Conditional Operators<br/>= Mote and to<br/>= Mote and</th> <th>The for LoopExtForm:Exterment;Extstatement;statement;forfor (initialization; test; update)forstatement;statement;statement;statement;statement;statement;form:statement;statement;statement;statement;statement;form:case integer-constructform:case integer-constant:case integer-constant:coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutform:statement(s);form:case integer-constant:case integer-constant:coutstatement(s);coutstatement(s);forminstatement(s);forminformonly used stream manipulatorsNameoutut to the beginningfixedsets fixed point notationleftsets fixed point notation</th> <th><pre>ample:<br/>(count &lt; 10; count++)<br/>(count = 0; count &lt; 10; count++)<br/>(count = 0; count &lt; 10; count++)<br/>cout &lt;&lt; "The value of count is ";<br/>cout &lt;&lt; end1;<br/>cout &lt;&lt; end1;<br/>cout &lt;&lt; end1;<br/>cout &lt;&lt; "The value of count is ";<br/>cout &lt;&lt; end1;<br/>cout &lt;</pre></th> | Commonly Used Operators<br>Assignment Operators<br>Assignment Operators<br>Assignment Operators<br>= assignment<br>= combined subtraction/assignment<br>= Combined mulpication/assignment<br>= Combined mulpication/assignment<br>= Combined modulo/assignment<br>= Subtraction<br>= Mulpication<br>= Mulpication<br>= Mulpication<br>= Less than or equal to<br>= Less than or equal to<br>= Less than or equal to<br>= Conditional Operators<br>= Mote and to<br>= Mote and | The for LoopExtForm:Exterment;Extstatement;statement;forfor (initialization; test; update)forstatement;statement;statement;statement;statement;statement;form:statement;statement;statement;statement;statement;form:case integer-constructform:case integer-constant:case integer-constant:coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutstatement(s);coutform:statement(s);form:case integer-constant:case integer-constant:coutstatement(s);coutstatement(s);forminstatement(s);forminformonly used stream manipulatorsNameoutut to the beginningfixedsets fixed point notationleftsets fixed point notation | <pre>ample:<br/>(count &lt; 10; count++)<br/>(count = 0; count &lt; 10; count++)<br/>(count = 0; count &lt; 10; count++)<br/>cout &lt;&lt; "The value of count is ";<br/>cout &lt;&lt; end1;<br/>cout &lt;&lt; end1;<br/>cout &lt;&lt; end1;<br/>cout &lt;&lt; "The value of count is ";<br/>cout &lt;&lt; end1;<br/>cout &lt;</pre> |
|---|---|---|--|
| <pre>ilse if Example<br/>(expression) if (x &lt; y)<br/>statement; x++;<br/>statement; n++;<br/>statement; x;<br/>se if (expression) else if (x &lt; z)<br/>x;<br/>se y++;<br/>conditionally-execute more than one<br/>ense, else<br/>statement; n-the statements in braces:<br/>im (expression) if (x &lt; y)<br/>f x++;<br/>statement; x++;<br/>statement; y ++;</pre>  | <pre>x = b;<br/>The while Loop<br/>Forn:<br/>While (expression) while (x &lt; 100)<br/>statement;    oout &lt;&lt; x++ &lt;&lt; endl;<br/>statement;</pre>  | <pre>right sets right justification set precision sets the number of significant digits set v sets field width showpoint forces decimal point &amp; trailing zeros to display Example: cout &lt;&lt; setprecision(2) &lt;&lt; fixed</pre>   | . width sets field width   |