



UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2007/2008 – 2<sup>nd</sup> Year Examination – Semester 4

**IT4103: Programming II**  
**PART 2 - Structured Question Paper**

6<sup>th</sup> September, 2008  
(ONE HOUR)

To be completed by the candidate

BIT Examination Index No: .....

**Important Instructions:**

- The duration of the paper is **1 (one) hour**.
- The medium of instruction and questions is English.
- This paper has **4 questions** and **8 pages**.
- **Answer all questions**
- **Write your answers** in English using the space provided **in this question paper**.
- Do not tear off any part of this answer book.
- Under no circumstances may this book, used or unused, be removed from the Examination Hall by a candidate.
- Note that questions appear on both sides of the paper.  
If a page is not printed, please inform the supervisor immediately.

**Questions Answered**

Indicate by a cross (x), (e.g. ) the numbers of the questions answered.

To be completed by the candidate by marking a cross (x).	Question numbers			
	1	2	3	4
To be completed by the examiners:				

**Read the following description and answer questions 1 - 3.**

One has to write a Java program to reverse the letters in a given word or sentence. When the program is executed it will prompt for entering a word or sentence to the user. Once the input is done as illustrated below:

PRADEEP KARIYAWASAME

and when the Enter key is pressed it should display the word or sentence with the letters in reverse order as follows:

EMASAWAYIRAK PEEDARP

In the program there is a one class to be used to implement the Stack. The stack should be implemented as an array having the basic functionality of a Stack assuming the name of the stack class as StackReverseWord. In the class two methods named push and pop should be implemented and other than that to check the top or the current usage of the Stack a method having the name peek should be implemented. Finally the class should have an ability to check whether the Stack is empty.

In order for the purpose of reversing the word or sentence one has to write another class with the name Reverse where the process of accepting the input word or sentence or outputting the reserved word or sentence should be taken place. The Reverse class directly interacts with the user and his or her input. Once the user made an input then the Reverse class accepts it and keeps them as a String. Then the input String is broken into single piece of characters in the Reverse class and sends them to the Stack one by one and kept them in the Stack array. When it is required to reverse the input word or sentence then the Reverse class interacts with the StackReverseWord class and takes the characters one by one from the Stack and the reversed word or sentence is formed in the Reverse class it self. Finally Reverse class is responsible for outputting the word or sentence with the letters in reverse pattern in the command prompt.

Finally one has to implement the reverse word or sentence process as a program in a separate class having the name DriverProgram. In the program a suitable user message should be displayed for user to start inputting the required string and when the user is pressed the Enter key after key after input then it should output the entered word or sentence with the letters in the reverse order.

- 1) Write the class definition for the StackReverseWord. ( 25 marks)
- a) Declare the required variables for the StackReverseWord considering the following instructions.
- A variable to indicate the maximum size of the Stack assuming that the maximum size is entered by the user.
  - A character array to maintain the Stack.
  - A variable to keep track of the top of the Stack to indicate the current usage of it

**ANSWER IN THIS BOX**

```
private int maxSize;
```

```
private char[] stackArray;
```

```
private int top;
```

- b) Write the constructor for the class StackReverseWord. In the constructor the variable holding the maximum size of the array has to be initialize based on the user's input. Then the array has to be created based on the value of the maximum size and the indicator of the top of the stack need to be initialized.

**ANSWER IN THIS BOX**

```
public StackReverseWord (int max)
```

```
{
```

```
    maxSize = max;
```

```
    stackArray = new char[maxSize];
```

```
    top = -1;
```

```
}
```

- c) Write the method which is responsible for pushing characters into the Stack one by one.

**ANSWER IN THIS BOX**

```
public void push(char j)
```

```
{
```

```
    stackArray[++top] = j;
```

```
}
```

d) Write the method which is responsible for popping characters from the Stack one by one.

**ANSWER IN THIS BOX**

```
public char pop()
```

```
{
```

```
return stackArray[top--];
```

```
}
```

e) Write the method which is responsible for identifying the top of the Stack.

**ANSWER IN THIS BOX**

```
public char peek()
```

```
{
```

```
return stackArray[top];
```

```
}
```

f) Write the method which is responsible for checking emptiness of the Stack.

**ANSWER IN THIS BOX**

```
public boolean isEmpty()
```

```
{
```

```
return (top == -1);
```

```
}
```

2) Write the class definition for the Reverse class.

(25 marks)

**ANSWER IN THIS BOX**

```
class Reverse
{
private String input;
private String output;

public Reverse (String in)
{ input = in; }

public String doRev()
{
int stackSize = input.length();
StackReverseWord theStack = new StackReverseWord (stackSize);
for(int j=0; j<input.length(); j++)
{
char ch = input.charAt(j);

theStack.push(ch);
}
output = "";
while( !theStack.isEmpty() )
{
char ch = theStack.pop();
output = output + ch;
}
return output;
}
}
```

3) Write the DriverProgram to implement the Reverse class.

**ANSWER IN THIS BOX**

```
import java.io.*;
class DriverProgram
{
public static void main(String[] args) throws IOException
{
String input, output;
while(true)
{
System.out.print("Enter a string: ");
System.out.flush();
input = getString();
if( input.equals("") )
break;

Reverse theReverse = new Reverse(input);
output = theReverse.doRev();
System.out.println("Reversed: " + output);
} }
public static String getString() throws IOException
{
InputStreamReader isr = new InputStreamReader(System.in);
BufferedReader br = new BufferedReader(isr);
String s = br.readLine();
return s;
}
}
```

4) Consider the following adjacency list which represents a Graph having 7 nodes in it.

a	c	d	f	
b	d	e		
c	a	f		
d	a	b	e	f
e	b	d		
f	a	c	d	
g				

a) Convert the above adjacency list in to an adjacency matrix to represent the same Graph.

(08 marks)

**ANSWER IN THIS BOX**

	a	b	c	d	e	f	g
a	0	0	1	1	0	1	0
b	0	0	0	1	1	0	0
c	1	0	0	0	0	1	0
d	1	1	0	0	1	1	0
e	0	1	0	1	0	0	0
f	1	0	1	1	0	0	0
g	0	0	0	0	0	0	0

b) Convert the above adjacency list in to an incidence matrix to represent the same Graph.

(08 marks)

**ANSWER IN THIS BOX**

	ac	ad	af	bd	be	cf	de	df
a	1	1	1	0	0	0	0	0
b	0	0	0	1	1	0	0	0
c	1	0	0	0	0	1	0	0
d	0	1	0	1	0	0	1	1
e	0	0	0	0	1	0	1	0
f	0	0	1	0	0	1	0	1
g	0	0	0	0	0	0	0	0

c) Convert the above adjacency list into a Graph having required number of nodes and vertices.

(09 marks)

**ANSWER IN THIS BOX**

\*\*\*\*\*