UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2007/2008 –2nd Year Examination – Semester 4

IT4103: Programming II

Part 1: Multiple Choice Question Paper

6th September, 2008
(ONE HOUR)

Important Instructions :

- The duration of the paper is **1 (one) hour.**
- The medium of instruction and questions is English.
- The paper has **25 questions** and **7 pages**.
- All questions are of the MCQ (Multiple Choice Questions) type.
- All questions should be answered.
- Each question will have 5 (five) choices with **one or more** correct answers.
- All questions will carry equal marks.
- There will be a penalty for incorrect responses to discourage guessing.
- The mark given for a question will vary from 0 (*All the incorrect choices are marked & no correct choices are marked*) to +1 (*All the correct choices are marked & no incorrect choices are marked*).
- Answers should be marked on the special answer sheet provided.
- Note that questions appear on both sides of the paper.
  If a page is not printed, please inform the supervisor immediately.
- Mark the correct choices on the question paper first and then transfer them to the given answer sheet which will be machine marked. **Please completely read and follow the instructions given on the other side of the answer sheet before you shade your correct choices.**

1) Select from among the following, the valid statement(s) on Recursions.

(a) Recursion is a programming technique in which a method or a function is called by itself. (b) Recursion is an Abstract Data Type. (c) A recursion consists of one Linked List object and a number of Linked objects. (d) A Recursion is a data structure which helps very fast insertion and deletion of data. (e) The shortest path of a Graph data structure has been created based on recursions.

Consider the following program written in Java to answer questions 2 and 3.

```
class What{
      public static void what(int i) {
       if (i>0){
        System.out.println(i + "");
       what(i-1);
        }
      }
      public static void main(String args[]){
        What.what(3);
        }}
```

2) Select from among the following, the correct recursive type which has been used in the program.

(a) Nontail	(b) Tail	(c) Indirect
(d) Nested	(e) Excessive	

3) What would the output of the program be?

(a) 123	(b) 321	(c) 6
(d) 1 2 3	(e) 3 2 1	

4) One has written a Java program, saved it with the name *MyFirstApplet.java* and complied successfully. After the following code segment has been written in a separate Notepad, one has saved it with the name *MyFirst.htm*.

```
<applet code="MyFirstApplet.class" width=300 height=300> </applet>
```

Select from among the following, the correct option(s) to execute the above file.

(a) javac MyFirstApplet.class	(b) appletviewer MyFirstApplet.htm
(c) appletviewer MyFirstApplet.java	(d) javac MyFirstApplet.java
(e) appletviewer MyFirst.htm	

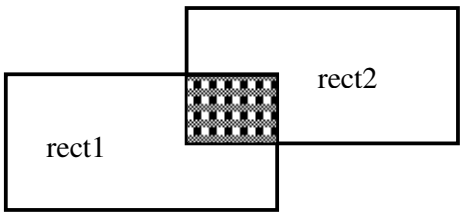
5) Select from among the following, the valid statements(s) which is/are illustrating the relative path of file access.

(a) (“dir” + File.separator + “output.txt”);	(b) (“C:” + File.separator + “MyFiles”);
(c) (“output.txt”);	(d) (\\MyFolder\\JavaFiles\\Files,”Java.java”);
(e) (“C:/MyFiles/Files”);	

6) Select from among the following, (a) valid method(s) which is/are available in a File object.

(a) getName()	(b) getPath()	(c) equals()
(d) toString()	(e) stop()	

7) Consider the following two rectangles having the names *rect1* and *rect2* taking note of the shaded area.



Select from among the following, the correct rectangle combining method available in the Rectangle class which can be used to get the highlighted area as a separate rectangle.

(a) rect1.union(rect2)	(b) rect1.intersection(rect2)	(c) rect1.add(rect2)
(d) rect1.grow(rect2)	(e) rect1.add(x, y)	

8) Consider the following description on a layout manager which is available in Java.

"Places components in the container in a rectangular grid with the number of rows and columns specified by the programmer"

Select from among the following, the correct layout manager which matches with the above description.

(a) FlowLayout	(b) Borderlayout	(c) SpringLayout
(d) GridLayout	(e) BoxLayout	

9) Select from among the following, the event listener interfaces which can be considered as Semantic event listeners.

(a) WindowListener	(b) ActionListener	(c) FocusListener
(d) ItemListener	(e) KeyListener	

10) Consider the following two columns A and B. In the column A, names of components which have some connection with Java database connectivity is written and in column B, descriptions of such components are given.

Column A	Column B
1 Driver manager	A A SQL statement to perform query or update operation
2 Statement	B Information about returned data, the database and the driver
3 Metadata	C Loads database drivers and manages the connection between the application and the driver.
4 Resultset	D Logical set of columns and rows of data returned by executing a statement

Select from among the following, the correct option which matches each component name in the column A with the appropriate description in the column B.

(a) 1 → A 2 → B, 3 → C 4 → D	(b) 1 → B 2 → A 3 → C 4 → D
(c) 1 → A 2 → B 3 → D 4 → C	(d) 1 → D 2 → B 3 → C 4 → A
(e) 1 → C 2 → A 3 → B 4 → D	

11) Select from among the following, the correct option(s) which can be used to load ODBC (Open Database Connectivity) driver for database connectivity in Java.

(a) Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
(b) Class.nameFor("sun.odbc.jdbc.JdbcOdbcDriver");
(c) Class.nameFor("sun.odbc");
(d) Class.nameFor("jdbc.JdbcOdbcDriver");
(e) Class.forName("JdbcOdbcDriver");

12) Select from among the following, the correct statement(s) on Linked Lists.

(a) A linked List consists of one Linked List object and a number of link objects.
(b) The link object contains a reference, often called last, to the first link in the list.
(c) Each Linked List object contains data and reference, often called next, to the next Linked List in the list.
(d) If the value null is stored in the next reference, it is an indication of the end of the list.
(e) In a doubly-Linked list, each link contains a reference to the previous link as well as the next link.

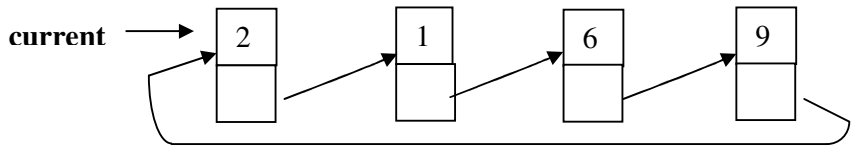
13) Select from among the following, the correct statement(s) on stacks.

(a) A stack allows access to only one data item at a time.
(b) In a stack, the last item inserted can be accessed first.
(c) It is a storage mechanism of a stack defined to be Last-In-First-Out(LIFO).
(d) Stack data structure can be implemented by other kinds of storage structures such as arrays.
(e) In stack terminology, the word *push* refers to taking an item from the stack.

14) Select from among the following, the correct statement(s) on the different types of Hash functions.

(a) There are two types of folding hash function, shift folding and boundary folding.
(b) In the mid-square function, the key is squared and the middle or mid part of the result is used as the address.
(c) In the extraction hash function TSize = sizeof(table), as in h(K) = K mod TSize is used if K is a number.
(d) Only a part of the key is used for a division hash function.
(e) The key value is transformed to another number base in the division hash function.

15) Consider the following diagram illustrating a kind of a linked list data structure.



Select from among the following, the name which can be given for the above illustration.

(a) Single Linked List
(b) Double Ended Linked List
(c) Circular Singly Linked List
(d) Doubly Linked List
(e) Skip List

16) Consider the following program written in Java to illustrate a Linked List.

```
public class ClassWhat{
 public int infor;
 public ClassWhat next;
 public ClassWhat(int k, ClassWhat j){
     infor = k;
     next = j;
  }}
```

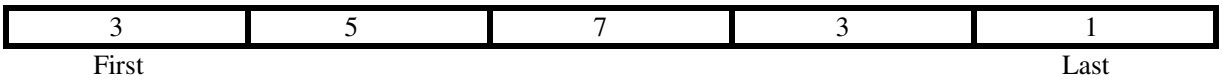
Select from among the following, the correct statement(s) about the program.

(a) The ClassWhat includes two data fields having the names infor and next.
(b) The infor field can be called an instance variable.
(c) The field next cannot be considered as a class variable.
(d) The field next can be termed as a self referenced object.
(e) The field next is indispensable for the implementation of the Linked List.

17) Single Linked lists have one serious drawback since it is required in sequential scanning to locate the desired value. Select from among the following, the Linked List type which can be used to eliminate the said drawback.

(a) Stack	(b) Double Ended Linked List
(c) Circular Singly Linked List	(d) Doubly Linked List
(e) Skip List	

18) Consider the following diagram illustrating a Queue data structure formed as an array. The values are also assigned to the array and the first element is marked with the name First and the last element with the name Last for clarity.



After creating the Queue, the following series of operations is performed on it.

enqueue(8)
dequeue()
dequeue()
enqueue(5)
dequeue()
dequeue()
dequeue()
dequeue()
clear()
enqueue(4)
enqueue(3)
enqueue(1)
enqueue(2)
enqueue(1)
dequeue()
enqueue(1)

Select from among the following, the content of the Queue after those operations.

(a) 35731	(b) 35738	(c) 43121
(d) 31211	(e) 43111	

19) Select from among the following, the package(s) which can be used for event handling in Java.

(a) javax.swing.*	(b) java.event.*	(c) java.io.*
(d) java.util.*	(e) java.awt.*	

Consider the following two classes written in Java on two Notepads and answer questions 20 and 21

```
public class ClassTreeWhat{
      protected int key;
      protected ClassTreeWhat left, right;
      public ClassTreeWhat(){
                  left = right = null;
       }
public ClassTreeWhat(int el){
    this(el,null,null);
          }

public ClassTreeWhat(int el,ClassTreeWhat lt, ClassTreeWhat rt){
    key =  el;  left = lt; right = rt;
          }
}

public class ClassTreeHow{
protected ClassTreeWhat root;
public ClassTreeHow(){
   root = null;
  }
}
```

20) Assume that one has written the following method in the class ClassTreeHow.

```
public void whatAlgorithm(int el){
  ClassTreeWhat p = root, prev = null;
   while( p != null){
        prev = p;
        if(p.key < el)
              p = p.right;
       else p = p.left;
      }
    if (root == null)
         root = new ClassTreeWhat(el);
    else if(prev.key < el)
        prev.right = new ClassTreeWhat(el);
    else prev.left = new  ClassTreeWhat(el);
       }
```

Select from among the following, the objective of writing the method in the class.

(a) To locate an element	(b) To insert an element	(c) To modify an element
(d) To delete an element	(e) To shift an element	

21) In order to find the minimum spanning tree of a Graph data structure, a number of algorithms can be referred to. Select from among the following, (an) algorithm(s) which can be used to solve the problem of finding a minimum spanning tree.

(a) Kruskal	(b) Ford	(c) Label Correcting
(d) WFI	(e) Dijkstra	

22) Assume that one has implemented the following pseudo code in the class ClassTreeHow.

```
algorithmWhat()
  while not finished
   if node has no left descendants
        visit it;
       go to the right;
else make this node the right child of the rightmost node in the left descendant;
       go to this left descendants;
```

Select from among the following, the objective of writing the method in the class.

(a) To shift an element	(b) To insert an element	(c) To modify an element
(d) To delete an element	(e) To search for an element	

23) Select from among the following, a limitation which exist in Tree Data structures which **cannot** be seen in Graphs.

(a) Tree data structures can only represent relations of hierarchical type, such as relation between a parent and a child whereas, in Graphs, one can represent relations of any kind.
(b) In Tree data structures, one can see nodes and connection in between them but in Graphs, one cannot see nodes or connections in between them.
(c) In Tree data structures, one can store only numbers as opposed to Graphs where any type of data is possible for storage.
(d) In Tree data structures, one cannot search for a value but in Graphs, searching for a value is possible.
(e) One cannot implement a Tree data structure in Java but a Graph can be implemented in Java.

24) Consider the following segment of program written in Java.

```
public int whatSort(int[] data) []{
 for(int i = 1 , j ; i  <= data.length ; i++){
     int tmp = data[i];
  for ( j = i   ;  j > 0 && tmp < data[j-1]  ;  j++)
          data[j]   =   data[j-1];
  data[j] = tmp;
       }
return data;
   }
```

Select from among the following, the name of the sorting algorithm which has been implemented in the program.

(a) Insertion	(b) Selection	(c) Bubble
(d) Shell	(e) Merge	

25) Select from among the following, what can be considered as sorting algorithms.

(a) Insertion	(b) Quick	(c) Bubble
(d) Radix	(e) Merge	

****