Q.6 Diagrammatically represent the following scenario of inheritance:

- We have **Shape** as a base class and **Circle, Rectangle, Triangle** and **Square** as its derived classes.

Q.7 Write a C++ program implementing a class with the name **ConstDest** having constructor and destructor functions in its body.

Q.8 Write a C++ program implementing a class with the name **Time.** This class should have a constructor to initialize the time, **hours, minutes** and **seconds**. The class should have another function named **ToSecond()** to convert and display the time into seconds.

Q.9 What is object? How objects are created to access members of a class?

Q.10 Explain inheritance and polymorphism with examples.

UNIT 9

FILE HANDLING

After the completion of Unit-9, Students will be able to:

- Know the binary and text file
- Open the file in different modes
- Know the concept of
  - bof()
  - eof()
- Define stream
- Use the following stream
  - Single character
  - String

INTRODUCTION

A **file** is a collection of related records that are permanently stored in secondary storage. File handling is an important feature of C++ language. This unit describes different types of files such as text files and binary files and describes how different operations like opening, reading, writing and closing are performed on files.

9.1 File handling

The combinations of characters, words and records are called **files.** The operations on these files are the major focus point of file handling. Suppose we have a file on the disk and want to open it then reading from or writing into the file before closing it will be termed as **handling files**. The basic steps involved in file handling are:

- Opening file
- Reading and writing a file
- Closing file

9.1.1 Types of files

C++ divides files into two different types based on how they store data. These are:

- Text files
- Binary files

Text files are those files that store data in text format which is readable by humans. Example of text file is the C++ source program which can be read by human.

Binary files store data in binary format which is not readable by the humans but readable by the computers. Binary files are directly processed by the computers. Binary files are normally used to store large data files. They take less space to store data as compared to text files. For example, an integer having length of six digits will take six bytes to accommodate in a text file while in binary files they will take only two bytes. The best example of binary file is the *object file* of a C++ source file that is generated by the compiler of C++.

9.1.2 Opening file

To open a file, the function **open ()** is used whose syntax is:

myFile.open(filename);

Here, **myFile** is an internal variable, actually an object, used to handle the file whose name is written in the parenthesis. To declare this variable the following statement is used:

Ifstream myFile;

The dot . operator is used between the variable *myFile* and *open* function. *myFile* is an object of *ifstream* while *open*() is a function of *ifstream*. The argument for *open* function is the name of the file on the disk which should be enclosed in double quotes. The file name can be simple file name like "sale.txt". It can also fully qualify path name like "C:\myprogs\sale.txt".

a. Modes of opening a file

While opening a file, we tell the compiler what we want to do with it i.e. we want to read the file or write into the file or want to modify it. In order to open a file in any desired mode the member function **open** () should take mode as an argument along with the file name. Its general syntax is shown below:

open (filename, mode);

Here, file name representing the name of the file to be opened, and mode is an optional parameter with a combination of the following flags:

ios::in	Open for input operations.
ios::out	Open for output operations.
ios::binary	Open in binary mode.
ios::ate	Set the initial position at the end of the file. If this flag is not set to any value, the initial position is the beginning of the file.
ios::app	All output operations are performed at the end of the file, appending the content to the current content of the file. This flag can only be used in streams open for output-only operations.
ios::trunk	If the file opened for output operations already exists then its previous contents are deleted and replaced by the new one.

All these flags can be combined using the bitwise operator OR |. For example, if we want to open the file **test.bin** in binary mode to add data we could do it by the following call to member function **open**().

```
Ofstream myfile;
myfile.open ("test.bin", ios::out | ios::app | ios::binary);
```

Each one of the **open**() member functions of the classes *of stream*, *ifstream* and *fstream* has a default mode that is used if the file is opened without a second argument:

class	default mode parameter
ofstream	ios::out
ifstream	ios::in
fstream	ios::in | ios::out

For *ifstream* and *ofstream* classes, ios::in and ios::out are automatically and respectively assumed, even if a mode that does not include them is passed as second argument to the open() member function.

Let us see an example program that creates a text file "**example1.txt**" and writes a line of text in it.

```
//opening a file and writing into it
#include <iostream.h>
#include <fstream.h>
#include <conio.h>
int main()
{
Ofstream myfile;
```



```
myfile.open ("example1.txt");
myfile<< "This is a program that tells you how to write to a file.\n";
myfile.close();
getch();
return 0;
}
```

This code creates a file called *example1.txt* and inserts a sentence into it in the same way as we do with *cout*, but using the file stream *myfile* instead. Figure 9.1 shows the output of the above program in the form of a text file that is created in the same working directory.

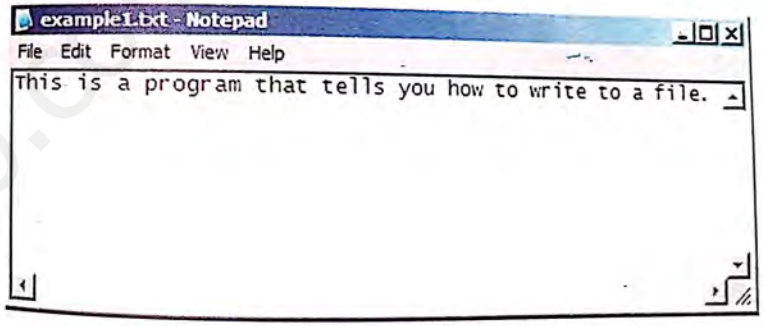


Figure 9.1: Opening and Writing into a Text File

b. Reading input file

To read a word from the file we can write as:

```
myFile>> c;
```

So, the first word of the file will be read in c, where c is a character array. It is similar as we used with **cin**. There are certain limitations to this. It can read just one word at one

time. It means, on encountering a space, it will stop reading further. Therefore, we have to use it repeatedly to read the complete file. We can also read multiple words at a time as:

myFile>> c1 >> c2 >> c3;

The first word will be read in c1, second in c2 and third in c3. Before reading the file, we should know some information regarding the structure of the file. If we have a file of an employee, we should know that the first word is employee's name, second word is salary etc,so that we can read the first word in a **string** and second word in an **int** variable.

Consider the following input file "*inputfile.txt*", shown in figure 9.2, which is read and displayed on the screen.

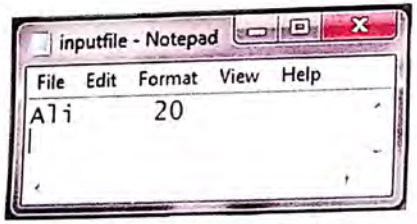


Figure 9.2: Input File to be Read

```
//reading input file program
#include <iostream.h>
#include <fstream.h>
#include <conio.h>
int main()
{
Ifstream myfile("c:\\inputfile.txt");
Char ch [20];
int m;
myfile>>ch>>m;
cout<<ch<<"\t"<<m;
myfile.close();
getch();
return 0;
}
```

The output of the above program is shown in figure 9.3:

Figure 9.3: Output of Reading Program

Let us write another simple program, to read from a file '**myfile.txt**' that is in the current directory, and print it on the screen. "myfile.txt" contains employee's name, salary and department in which they are employed. Figure 9.4 shows **myfile.txt** which is followed by the complete program to describe how it is opened, read and closed.

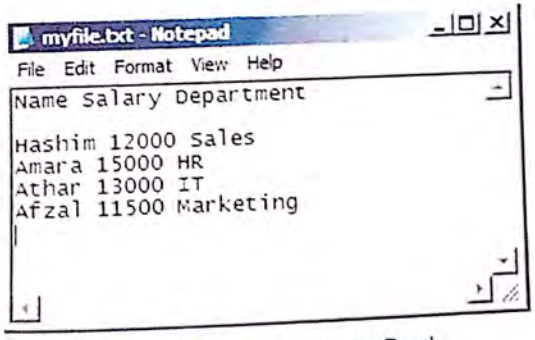


Figure 9.4: Input File to be Read

```
#include <iostream.h>
#include <fstream.h>
#include <conio.h>
main()
{
Char name[50]; // used to read name of employee from file
Char sal[10]; // used to read salary of employee from file
Char dept[30]; // used to read dept of employee from file
Ifstream inFile; // Handle for the input file
Charin putFileName[] = "myfile.txt"; // file name, this file is in the current directory
inFile.open(inputFileName); // Opening the file
// checking that file is successfully opened or not
if (!inFile)
{
cout<< "Can't open input file named " <<inputFileName<<endl;
exit(1);
}
// Reading the complete file word by word and printing on screen
while (!inFile.eof())
{
inFile>> name >>sal>>dept;
cout<< name << "\t" <<sal<< " \t" <<dept<<endl;
}
inFile.close();
getch();
return 0;
}
```

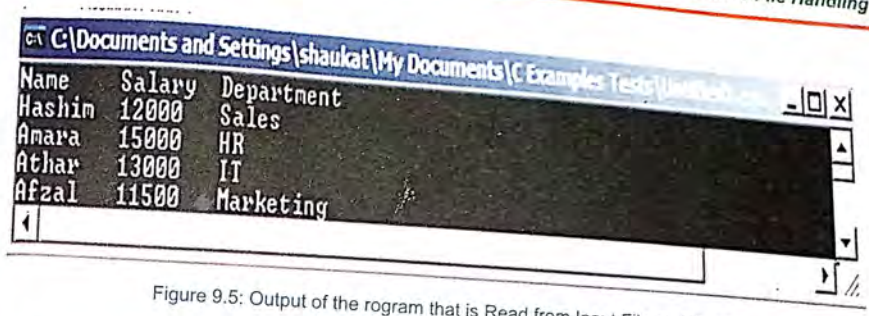



Figure 9.5: Output of the rogram that is Read from Input File myFile.txt

c. Closing file

Once we have read the file, it must be closed. It is the responsibility of the programmer to close the file. We can close the file by using the following statement:

myFile.close();

The function **close**() does not require any argument, as we are going to close the file associated with **myFile**. Once we close the file, no file will be associated with **myfile**.

d. Opening files in binary mode

To open a file in binary mode, we need to set the file mode to **ios::binary**. Suppose we have a binary file named as "**test.dat**". To open this file in binary mode, we write the statement as:

ofstream myFile;
myFile.open ("test.dat", ios::binary);

In order to write the data to a binary file, "**write**" method is used. This method is a member function of **ofstream** or **fstream** class.

9.1.3 bof() and eof()

C++ provides special functions **bof()** and **eof()** that are used to set the pointers to the beginning of a file and at the end of a file respectively. The following sections explain them with the help of proper examples.

a. bof() – beginning-of-file

The **bof()** is a pointer which returns true if the current position of the pointer is at the beginning of the input file stream, and false otherwise. It means that it tells the compiler whether the cursor is at the beginning of file or not.

myFile.close();

b. eof() – end-of-file

The **eof()** is a pointer which returns true when there are no more data to be read from an input file stream, and false otherwise. It means that this function checks whether control has reached to the end of file or not. This function is very useful in the case when we do not know the exact number of records in a file.

Rules for using eof( )

- Always test for the end-of-file condition before processing data.
- Use a **while** loop for getting data from an input file stream. A **for** loop is desirable only when you know the exact number of data items in the file

```
// reading a text file using eof() function
#include <iostream.h>
#include <conio.h>
#include <fstream.h>
int main()
{
char ch [50];
ifstream flag ("c:\\TestRecord.txt");
```



```
cout<<"Output is"<<endl;
while(!flag.eof())
{
flag>>ch;
cout<<ch<<endl;
}
flag.close();
getch();
return 0;
}
```

Output of the program

```
Output is
Ali      20
Anwar 15
Zainab 17
Zonash21
```

In the above program, the input file *TestRecord* is loaded and the *eof ()* function detects the end of file. The program reads the file record by record into the string variable *ch* which is then displayed on the screen.

9.1.4 Stream

A **stream** can be thought of as a sequence of bytes of infinite length that is used as a buffer to hold data to be processed. In C++ a **stream** is a sequence of bytes associated with a file. Most of the times streams are used to assure a good and secure flow of data between an application and file.

In C++ there are two types of streams, ***input stream*** and ***output stream***.

Input streams take any sequence of bytes from an input device such as a keyboard, a file, or a network, while **output streams** are used to hold output for a particular data consumer, such as a monitor, a file, or a printer. When writing data to an output device, the device may not be ready to accept that data e.g. the printer may still be warming up when the program writes data to its output stream. The data will reside in the output stream until the printer starts consuming it.

The act of taking data from the input devices is called *extraction* and the operator used for this purpose is called *stream extraction operator* >>. Similarly, the act of displaying data on the screen is called *insertion* and the operator used for this purpose is called *stream insertion operator* <<.

Sometimes, both operations **input/output** may need to be used at the same time. **Standard streams** in C++ consist of four predefined standard stream objects. These are: **cin, cout, cerr** and **clog.** C++ language handles files using these streams objects. For this purpose, the header file <**fstream.h**> is needed to be included.

If the file is only used for reading purpose then the header file **ifstream** (input file stream) is needed to be included. Similarly, if one wants to write in some file then header file **ofstream** (output file stream) is needed. One can read, write and manipulate the same file by using the header file **fstream**.

Consider the following example of input and output using the standard streams:

```
// input/output stream example
#include <conio.h>
#include <iostream.h>
#include <process.h>
int main()
{
float height;
cout<< "Enter your height: " <<endl; // output stream
cin>> height; // input stream
if (height <= 0)
{
cerr<< "You entered an invalid height!" <<endl;
exit(1);
}
cout<< "Your height is " << height << " feet" <<endl;
getch();
return 0;
}
```

Output of the program

```
Enter your height:5.8
Your height is 5.8 feet
```


9.1.5 Use of the Streams

The data can be read from and written to files with the help of single character stream and string stream.

a. Single character stream

Using single character stream, the data can be read from and written to files character by character.

i. Reading files character by character

The function *get()* is used to read data character by character from files.

Consider the following program that reads the data one character at a time from the file "*characters file.txt*",shown in figure 9.7, and display them on the screen.

```
//character stream (read) program
#include <conio.h>
#include <iostream.h>
```

```
#include <fstream.h>
int main()
{
char ch;
ifstream reads("c:\\charactersfile.txt");
while(!reads.eof())
{
read.get(ch);
cout<<ch<<endl;
}
read.close();
getch();
return 0;
}
```

Output of the program

```
r
e
a
d

t
h
e

f
i
l
e

p
l
e
a
s
e
.
```

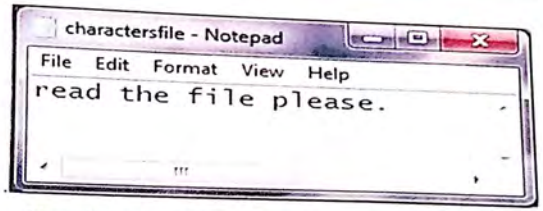



Figure 9.7: Input File to be Read Character by Character

ii. Writing files character by character

Similarly, using single character stream, data can be written to files character by character by using the function *put ()*. Consider the following program that writes data character by character to an empty file "*characterswrite.txt*".

```
//character stream ( write) program
#include <conio.h>
#include <iostream.h>
#include <fstream.h>
int main()
{
char ch;
ofstream writes("d:\\characterswrite.txt");
for (int i=0; i<=20; ++i)
{
cin>>ch;
cout<<writes.put(ch);
}
writes.close();
getch();
return 0;
}
```

When the above program is executed, characters are entered one by one from the keyboard.The characters entered above are written to the "*characterswrite*" file stored at secondary storage at location *d-drive* of the hard disk. The output file generated is shown in figure 9.8.

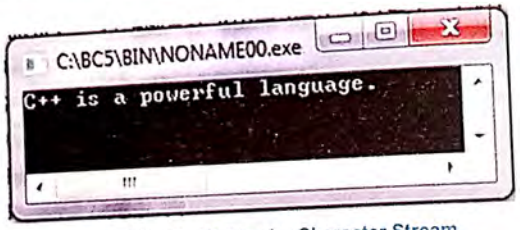


Figure 9.8: File Written by Character Stream

b. String stream

A **string stream** is a stream which reads input from or writes output to an associated string.Consider the text file "*stringread*" shown in figure 9.9.

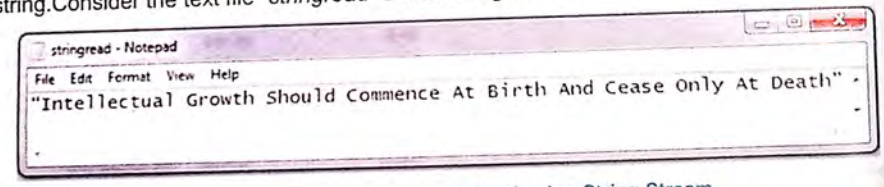


Figure 9.9: Input File that will be Read using String Stream

The following program reads this file (*stringread.txt*) into string (*str*) using *getline()* function and display the result on the screen as shown in figure 9.10.

```
//string stream (read) program
#include <conio.h>
#include <iostream.h>
#include <fstream.h>
#include <string>
int main()
{
char str[20];
```



```
ifstream reads("d:\\stringread.txt");
while(!reads.eof())
{
reads.getline(str,21);
cout<<str<<endl;
}
reads.close();
getch();
return 0;
}
```

Figure 9.12: File Read by String Stream

Summary

- The combinations of characters, words and records is called files.
- The process of opening, reading from and writing into files is termed as handling files.
- There are two types of files: text files and binary files.
- Those files that store data in text format and are readable by human are called text file.
- Binary files are those files that store data in binary format which is not readable by the human but readable by the computers. Binary files are directly processed by the computers
- To open a file, the function open() is used.
- There are different modes of opening files i.e. binary mode, input mode and output mode.
- The bof( ) is a pointer which is true if the current position of the pointer is at the beginning of the input file stream, and false otherwise.
- The eof( ) is a pointer which returns true when there are no more data to be read from an input file stream, and false otherwise.
- A stream can be thought of a sequence of bytes of varying length that is used as a buffer to hold data that is waiting to be processed

Exercise

Q.1 Fill in the Blanks.

i. In C++ language, for handling of files, the header file _____ is used.
ii. The function _____ is used for closing a function.
iii. In the statement, open ("xyz.txt", ios::out); _____ file is opened in output mode.
iv. If all output operations are to be performed at the end of the file then _____should be used in the open function.
v. A _____is "a continuous flow or succession of bytes a stream of characters."

Q.2 Select the correct choice for the following Multiple Choice Questions.

i. eof stands for _____.
   a. errors-on-files
   b. end-of-file
   c. exit-of-file
   d. none of the above
ii. In file handling open() function is used _____.
   a. to open C++ compiler
   b. to open a file
   c. both a and b
   d. none of the above
iii. Default mode parameter for ofstream is_____.
   a. ios::out
   b. ios::in
   c. ios::binary
   d. both a and b

iv. A text file has the extension_____.

a. .doc
b. .docx
c. .txt
d. .bin

v. ios::binary is used as an argument to open() function for_____.

a. opening file for input operation.
b. opening file for output operation.
c. opening file in binary mode.
d. none of the above.

Q.3 Write TRUE/FALSE against the following statements.

i. A string stream is a stream which reads input from or writes output to an associated string.
ii. The eof( )is a pointer which returns false when there are no more data to be read from an input file stream, and true otherwise.
iii. To open a file in binary mode, we need to set the file mode to ios::binary.
iv. close() Function is used in C++ file handling for opening a new file.
v. Those files that store data in text format and are readable by human are called text file

Q.4 Define file. Describe different types of file.

Q.5 Describe different types of operations performed on the files.

Q.6 Define streams. Describe input and output streams in detail.

Q.7 What is meant by the term mode of file opening? Describe different modes of opening file.

Q.8 What is file handling? Explain it in detail.

Bibliography

1. *Basic Computer: Operating System Functions.* [Cited 2011 Jun 10]; Available from: http://www .comptechdoc.org/basic/basictutl.
2. Stalling, W., *Operating Systems: Internals and Design Principles.* Sixth ed. 2009: Macmillan, New York.
3. *Operating System.* Webopedia (N.D) [cited; Available from: http://www .webopedia.comrERMIO//{)perating system html.
4. Silberschatz, Galvin, and Gagne, *Operating System Concepts.* Seventh ed: Addison-Wesley Inc.
5. Ritchie, C., *Operating System: Incorporating Unix and MS-DOS.* 2nd ed. 1997: BPB Publication, Delhi. 232.
6. *Common Operating Systems: Microsoft DOS.* (N.D) [cited 2011 May, 20th] Available from: http://www.wiley.com/college/busin/cmis/oakroan/outjine/chap05/misc/dos.htm.
7. *DOS (Disk Operating System).* 2004-04-03 [SB] Last Updated: 2010-05-14 [cited 2011 April, 20th]; Available from: http://www.operating- system.org/betriebssystem/ englishlbs-msdos.htm.
8. *Unix Tutorials: Introduction to the UNIX Operating System* 19 October 2001 [cited 2011 April 17]; Available from: http://www.ee.surrey.ac.ukffeaching/Unixlunixintro.html.
9. *History of Mac OS,* in *Wikipedia.* 2008.
10. *Common Operating Systems: Macintosh System 7.5 (Apple).* [cited 2011 March 28]; Available from: http://www.wiley.com/college/businlcmis/oakman/outline/chap05/misc/mac.htm.
11. *Harvey M. Deitel, Paul J. Deitel, and D.R. Choffues, Operating Systems (3rd Edition). 3rd ed. 2004.*
12. *Multi Programming Operating System.* June 2, 2011 [cited 2011 Jun 10]; Available from: http //easy2te.ach. netloperating-.systemlmulti-programming-operating-system.
13. Stankovic, I.A., Real-Time Computing. 1992, Department of Computer Science, University ofMassacusetts.
14. Fry, G. *Real Time Systems.* [cited 2011 May 10]; Available from: http://cs-people. bu.eduJ gfry/realtime.html.
15. *Physical Oceanographic Real-Time System (PORTS®).* [Cited 2011 April 8];

Available from: http://tidesandcurrents.noaa.gov/ports.html.
16. Tanenbaum, A.S., *Modern Operating Systems.* 3rd ed. 2008 Prentice-Hall.
17. Blaise Barney and L. Livermore, *Introduction to Parallel Computing,* National Laboratory.
18. Alecu, F., *Operating System for Parallel Processing.* Studies in Informatics and Control, 2004. 13(2).
19. Tannenbaum and M.v. Steen, *Distributed Systems.* 2001: Prentice Hall.
20. *Operating Systems Services and Functions.* UzEE November 4, 2007 [cited 2011 March 2]; Available from: http://uzeeinc.wordpress.coml2007111/04/operating-systems-services-and- functions/
21. *Muhammad, R..B. Operating Systems Lecture Notes: System Components. [Cited 2011 Jun 01]; Available from:* http://www.personal.kent.edu/~rmuhamma/OpSystems/os.htmL
22. *What Does an Operating System Do?* 2009 [cited 2011 May 7]; Available from: http://computerspot.netlwhat -does-an-operating-system-do/.
23. *File management system,* in *Webopedia,* Webopedia.
24. Bhat, P.C.P. *Operating Systems/Input Output (10) management Lecture Notes* Volume,
25. Dean, T., Comptia Network+ 2009 in Depth. 2009. 767.
26. *Operating System Notes: Threads.* [Cited 2011 May 21]; Available from: http://www.personal.kent.edu/~rrnuhamma/OpSystemslMyos/threads.htm.
27. Joydip Kanjilal, Operating Systems: Coneeptsand Terminologies, 31 Jul 2006, [cited 12, Dee 2011]
28. Aggarwal, K.K., *Software Engineering:* Revised Edition ed. 2001: New age international publisher. 494.
29. Goldsmith, R.F. *Software development life cycle phases, iterations, explained step by step.* Available from: www.SearchSoftwareQuality.com.
30. Gary B. Shelly, Thomas J. Cashman, and H.J. Rosenblatt, *System Analysis and Design,* Eight ed. 2009.
31. Ankur Patel, Sagar Padaliya, and D. Pande. *SDLC Methodologies.* [cited 2011 Jun 12]; Available from: http://community.mis.temple.edu/fox20/about/.
32. bijayani. *SDLC.* 2010 [cited 2011 Jun, 11]; Available from: http://bijayani.wordpress.com/2010/02115/sdlc/.
33. Pressman, R., *Software Engineering: A Practitioner's Approach.* seventh ed. 2010: McGraw-Hill Higher Education.

34. Hemphill, M. *MIS for digital age: Systems Development Lifecycle.* February 4, 2004 [cited 2011 Jun, 02]; Available from: http://exonous.typepad.comimis/2004/02/systems develop.html.
35. Sommerville, *Software Enginering.* Eight ed. 2008: Addision Wesely Publisher Limited 864.
36. Wasson, C.S., *System analysis, design, and development: concepts, principles and Practices.1005* John Wiley & Sons Inc. 818.
37. Glossary of Computer Systems Software Development Terminology (8/95)
38. "What is C++." [cited May 28, 2011]; 2005, Available from: http://www.hitmill.comiprogrammingl cpp/whatiscpp .html.
39. Stair, and Ralph M t et al., *Principles of Information Systems.* Sixth ed.: Thomson Learning, Inc, 2003.
40. *Program.* [cited May 28, 2011]; Available from: http://www.webopedia.com/TERMIP/program.html.
41. *"C++ : Documentation: C++ Language Tutorial." [Cited May 2. 2011J: Available from:* http://www.cpluspius.com.
42. Alex, "The C++ Tutorial.", [cited April, 2011]; available from: http://www.leamcpp.com/. 2007.
43. "Types of Constants in C++." [cited May 19, 2011]; Available from: http://www.adiwebs.com/types-of-constants-in-cl.
44. Sripriya, R., "C++ Manipulators." 2007. [last updated lith Nov 2008]. [cited 2011 29th May]; Available from: http://www exforsvs.com/tutorials/c-plus-plus/c- manipulators.html.
45. "C++ lessons and Topics: FunctionX Tutorial", available from: http://www.functionx.wm/, Last updated: Monday, June l3, 201 L
46. Robert Lafore, "Object-Oriented Programming in C++", 4th Edition, Dee 29, 2001.
47. Nicolai M. Josuttis, "Object-Oriented Programming in C++", December 30, 2002.
48. Paul Deitel and Harvey M. Deitel, "C++ How to Program", 7th Edition. Aug 16, 2009.
49. Paul Deitel and Harvey Deitel, "C++ How to Program (8th Edition)". Mar 25, 2011.
50. "String in C++", [cited 04, Jan 2012], available from: http://anaturb.netiC/stringexapm.htm
51. Bjame Stroustrup's C++ Glossary, Modified, February 19, 2007, available from: http://www?.research.att.coml~bs/ glossarv.html

Answers

Unit 1

Q.1 Fill in the Blanks.

i	ii	iii	iv	v
Multiprocessing	Multiple Tasks	Multiprogramming	Exit	UNIX

Q.2 Multiple Choice Questions.

i	ii	iii	iv	v	vi	vii	viii	ix	x
b	c	c	a	b	b	d	a	b	a

Q.3 TRUE/FALSE statements.

i	ii	iii	iv	v	vi	vii	viii	ix	x
T	T	T	T	F	F	F	F	F	T

Unit 2

Q.1 Fill in the Blanks.

i	ii	iii	iv	v
Functional	Deployment	System Analyst	Programmer	Project management

Q.2 Multiple Choice Questions.

i	ii	iii	iv	v	vi	vii
c	d	a	a	b	a	d

Q.3 TRUE/FALSE statements.

i	ii	iii	iv	v
T	F	T	F	T

Unit 3

Q.1 Fill in the Blanks.

i	ii	iii	iv	v
0x	Conditional operator	True/False	Initialization	main( )

Q.2 Multiple Choice Questions.

i	ii	iii	iv	v
b	a	a	a	c

Q.3 TRUE/FALSE statements.

i	ii	iii	iv	v
F	F	F	T	F

Unit 4

Q.1 Fill in the Blanks.

i	ii	iii	iv	v
Enclosed within a block	The value returned by a conditional expression	Default	Exit	iteration

Q.2 Multiple Choice Questions.

i	ii	iii	iv	v	vi	vii	viii	ix	x
c	b	e	b	d	d	c	a	c	c

Q.3 TRUE/FALSE statements.

i	ii	iii	iv	v
F	T	T	T	F

Unit 5

Q.1 Fill in the Blanks.

i	ii	iii	iv	v
0	10	Multidimensional	String	float

Q.2 Multiple Choice Questions.

i	ii	iii	iv	v
b	a	d	c	c

Q.3 TRUE/FALSE statements.

i	ii	iii	iv	v
F	T	F	T	F

Unit 6

Q.1 Fill in the Blanks.

i	ii	iii	iv	v
One	Semicolon (;)	Throughout the program	Function overloading	Void

Q.2 Multiple Choice Questions.

i	ii	iii	iv	v
c	c	a	a	d

Q.3 TRUE/FALSE statements.

i	ii	iii	iv	v
T	F	F	F	T

Unit 7

Q.1 Fill in the Blanks.

i	ii	iii	iv	v
Address	int	Initialization	Reference	Dereference

Q.2 Multiple Choice Questions.

i	ii	iii	iv	v
a	c	a	c	b

Q.3 TRUE/FALSE statements.

i	ii	iii	iv	v
F	F	T	T	T

Unit 8

Q.1 Fill in the Blanks.

i	ii	iii	iv	v
Constructor	Inheritance	Code reusability	Class	Polymorphism

Q.2 Multiple Choice Questions.

i	ii	iii	iv	v
b	c	d	c	c

Q.3 TRUE/FALSE statements.

i	ii	iii	iv	v
F	T	T	F	T

Unit 9

Q.1 Fill in the Blanks.

i	ii	iii	iv	v
<fstream.h>	close	xyz.txt	Ios::app	stream

Q.2 Multiple Choice Questions.

i	ii	iii	iv	v
b	b	a	c	c

Q.3 TRUE/FALSE statements.

i	ii	iii	iv	v
T	F	T	F	T

Glossary

Actual parameters
Actual parameters are those parameters which appear in function calls.

Address
A memory location within computer that is used to store value.

Algorithm
A finite set of well-defined rules for the solution of a problem in a finite number of steps.

Allocate
To assign a resource to a process for performing a specific task.

Alphanumeric
A character set that contains letters, digits, and usually other characters such as punctuation marks.

Analysis phase
In this phase, the in-charge of the project team must decide if the project should go ahead with the available resources or not.

Analyst
Analyst is a profession al in the field of software development that studies the problems, plans solutions for them, recommends software systems, and coordinates development to meet business or other requirements.

Argument
A value passed to a function from the calling program, "main()" function.

Argument passing
The process of passing values to the formal parameters mentioned in the function declator.

Array
An n-dimensional ordered set of data items identified by a single name and one or more indices, so that each element of the set is individually addressable; e.g., a matrix, or table.

Assignment operator
The operator denoted by the symbol (=) which is used to assign values to variables.

Auxiliary storage
Storage devices other than main memory e.g., disks and tapes.

Base class
A class from which other classes are derived. Also called parent class or super class.

Batch
A group of records or data processing jobs brought together for processing or transmission.

Batch processing
A feature of operating system in which more than one records (a batch) are accumulated and processed after a regular interval of time.

Batch processing system
It is that type of operating system which collects jobs in batches before being processed by the CPU.

Binary file
Binary files store data in binary format which is not readable by the human being but readable by the computers.

Binary operator
An operator that operates on two operands at a time, such as +, /, &&, etc.

Black box testing
It is that type of testing which tests all possible combinations of end-user actions. It focuses on validation tests, boundary conditions, destructive testing, and security-related testing that helps to identify the vagueness and contradiction in functional specifications.

Block
A group of statements enclosed in curly braces, {}.

Block comment
These types of comments are represented by the symbols /* ..... */

Blocked (or waiting) State
A process is said to be in blocked state if it is waiting for some event to happen such as an Input/output completion before it can proceed.

bof()
The bof( ) is a pointer which returns TRUE if the current position of the pointer is at the start of the input file stream, and FALSE otherwise.

Bool
The built-in data type that can have either true or false value.

Break statement
It is used to make jump from one part of a program to another.

Bug
A fault in a program which causes the program to perform in an unintended or unanticipated manner.

Built-in functions
These are pre-compiled functions for some commonly used operations.

Built-in type
A data type directly provided by C++, such as int, double, and char.

Byte
A unit of memory that can hold a character of the C++ representation character set. The size of a byte is equal to 8 bits.

C++
An object-oriented high-level programming language.

C++ statement
A C++ statement is a simple or compound expression that can actually produce some effect.

Call-by-reference
Calling a function in such a way that to pass a reference to the function rather than a value

Call-by-value
A type of method used to pass arguments to a function in which a copy of the arguments is passed.

Char
A built-in data type used to declare variables of character type to store character constants.

cin
Stands for console input. It is used to get values form the users at run time.

Class
A feature of object oriented programming languages used to divide large size programs into smaller units which have their own data members and member functions.

Coding
The transforming of logic and data from design specifications (design descriptions) into a programming language.

Comment
The explanation added with statements. These are block comment /* ... */ and line comment //.

Compilation
Translating a program expressed in source language into object code.

Compiler
A computer program that translates programs expressed in a high-level language into their machine language equivalents.

Compound assignment operators
These operators are (+=, -=, *=, /=, %=).

Compound expression
Compound expression is an expression that involves more than one operator.

Compound statement
Sequence of statements enclosed in curly braces { ... }.

Computer language
A language designed to enable humans to communicate with computers e.g. C, C++, Java etc.

Computer program
A set of computer instructions written in any computer language that perform a specific task is called computer program.

Concatenation of strings
Joining two strings into a single string is called string concatenation.

Const
A key word used to declare constant identifier whose values cannot be changed during program execution.

Constant
Those objects in C++ whose values cannot be changed during program execution i.e. integer constant, character constants and string constant.

Constructor
A constructor is a special kind of class member function that is executed when an object of the class is instantiated.

Constructor
A special member function with the same name as its class and used to initialize objects of its class.

Constructor overloading
Using more than one constructors in the same class is called constructor overloading.

Context Switch
This refers to switching the CPU from one process to another by saving the state of the old process in the stack and executing the new process.

Continue statement
The continue statement provides a convenient way to jump back to the top of a loop earlier than normal, which can be used to bypass the remainder of the loop for an iteration.

Cout
A C++ object used to print values and strings on the screen.

Customer
Customer is an individual or an organization that is a current or potential buyer or user of the software product.

Data hiding
Data hiding is an important concept of C++ programming language in which the members of a class are protected against illegal access from outside the class.

Data member
Member of a class that can hold a value. Usually data members are called variables. A data member can be a static member or a non-static member.

Data type
Data type is the property of variables which tells the compiler that what type of data the variables will store.

Debugging
Determining the exact nature and location of a program errors, and fixing them.

Decision statement
It is a statement that causes the program to change the path of execution, sequence of execution, based on the value of an expression.

Declaration
An introduction of a name of a variable or function into a scope by specifying its type.

Default argument
A value specified for an argument in a function declaration, to be used if a call of the function doesn't specify a value for that argument.

Default constructor
Constructor requiring no arguments. Used for default initialization.

Derived class
A class that inherits features from one or more base classes.

Design
The process of defining the architecture, components, interfaces, and other characteristics of a system or component.

Design phase
That phase of software development life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements.

Desktop
Desktop is the essential component of a graphical user interface of an operating that is used for the quick and easy access of data and other computer resources.

Destructor
Member of a class used to clean up before deleting an object. Its name is its class' name prefixed by '~'.

Developer
A person, or group, that designs and/or builds ocuments configures the hardware software of computerized systems.

Device manager
The module of operating system used to manage input/output devices is called Input/output manager or device manager.

Disk
A storage device that includes one or more flat, circular plates with magnetic or optical surfaces on which information is stored.

Dispatch
The process of allocating CPU to a ready job, waiting in ready queue, is called dispatching.

Distributed operating
A distributed operating system is an operating system that manages a group of independent computers and makes them appear to the users as a single computer.

Documentation
The aids provided for the understanding of the structure and intended uses of an information system or its components, such as flowcharts, textual material, and user manuals.

DOS
DOS is the abbreviation for Disk Operating System which was the first operating system used for personal computers.

Double
Double-precision floating-point number.

Driver
Drivers are the software which are placed between the hardware layer and the application or operating system. These software make the hardware ready to be used by the application software and users.

Economic feasibility
Refers to the cost of the project.

Embedded software
Software that is part of a larger system and performs some of the requirements of that system; e.g., software used in an aircraft.

End user
A person that uses an information system for the purpose of data processing.

eof()
When users process files, they need to check for the end of file, eof, pointer to know whether the end of file has reached or not. If the end of file has reached, the function returns true and false otherwise.

exit() function
This function tells the program to quit running immediately.

Expression
Combination of operators and operands.

Feasibility study
Analysis and evaluation of a proposed project or system, to determine, whether it is technically, economically and operationally feasible within the estimated cost and time is called feasibility study.

File
A collection of related data (records) that is stored on secondary storage with a unique name.

File and I/O Management
This is the module of the Operating System that is responsible for creating and/or deleting files in the file system and managing the input and output of data in the file system.

File System
A File System refers to the arrangement in a secondary storage that is done by the Operating System for the purpose of data storage and retrieval.

Floating-point type
A floating point type can be either float, double, or long double. It is typically represented as a mantissa and an exponent.

Flow control statements
It is also called flow control statements, which allow the programmer to change the flow of sequence of execution of statements of a program by the processor.

Flowchart
A flowchart is a type of diagram that represents an algorithm or a process.

Formal parameters
These are those parameters which appear in function declarator or function prototype.

for-statement
For-statement is an iteration statement specifying an initializer, an iteration condition, an increment/decrement operation, and a controlled statement.

fstream
A file stream for input and output.

Function
A self-contained program that is used to perform a specific task and be invoked/called from a calling program by providing its arguments is called a function.

Function call
A function call is a statement in the calling function (e.g. main() function) to execute the code of the function.

Function declaration
The declaration of a function tells the compiler about the name, argument types, and return type of the function.

Function definition
Function declarator that includes the full body of the function is called function definition i.e. function declarator plus function body.

Function overloading
is a feature of C++ that allows us to create multiple functions with the same name, so long as they have different number or types of parameters.

Function prototype
A function prototype is that part of a function that tells the compiler the name of the function, the type of data returned by the function, the number of parameters the function expects to receive, the types of the parameters, and the order in which these parameters are expected. It does not contain the function body.

Functional requirement
Are those requirements of a software system which describe a function of a software system or of its component. It includes calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

getch ()
This is a standard library function defined in the header file <conio.h> and is used to wait for the user to input a character from the keyboard.

Global function
A function declared outside any function is called global function.

Global variable
The variables defined at the top of a program before the main() function are called global variables.

Graphical User Interface (GUI)
Graphical User Interface is the interface with graphical components; such as icon, menus, buttons and lists etc. that provide easy access to the information stored on the computers.

Header file
Files that hold declarations for pre-defined C++ objects are called header files. Thus, a header file acts as an interface between separately compiled parts of a program.

Hierarchical file system
A file system in which a directory can logically contain other directories.

if-statement
Statement selecting between two alternatives based on a condition. In if construct, the statements in the body of if are executed only if the condition is true and nothing is executed if it is false.

ifstream
A file stream for input defined inside the stream fstream.

Implementation
The process of translating a design into hardware components, software components, or both

Inheritance
The feature of C++ language in which new classes are derived from -existing classes is called inheritance.

Initialization
Assigning values to C++ variables and objects during its declaration time is called Initialization.

inline function
These are special function declared in the programs with the inline keyword to exploit the advantage of faster execution time by overcoming the problem of function call overhead.

Input stream
Input streams take any sequence of bytes from an input device such as a keyboard, a file, or a network.

Integer type
It is a group of data type used in C++ that occurs in one of its forms: short, int, or long. It defines integer variables that store integer constants.

iostream
It is a standard library file (header) or stream that can be used for both input and output in C++ programs.

Job
A unit of work for an operating system.

Kernel
This is the core of Operating System that is responsible for Memory and Processor Management in the system.

Line comment
Comment that starts with symbol (//) and ends with end-of-line.

Local function
A function defined within a function. Not supported by C++. Most often, the use of a local function is a sign that a function is too large.

Local variable
The variables defined within a block are called local variables.

Logical operator
There are three logical operators used in C++. These are (!, &&, !!).

long double
A float type of data that is used to define variables to store extended-precision floating point numbers.

long int
Integer data type that occupies 4 bytes of memory and is used to store large integer values.

Loop
An iterative statement that executes a statement or group of statements zero or more times is called loop. Examples are for-loop, while-loop and a do-while loop.

Macintosh
It is a series of operating systems having graphical user interfaces that have been developed by Apple Inc.

main program
A component of a C++ program from where the execution of the C++ program starts and calls other functions from.

Maintenance
Activities such as adjusting, cleaning, modifying, repairing equipment to assure performance in accordance with requirements.

Management
The process of organizing, coordinating and controlling the activities of a software development by the managers and executives in accordance with certain standard procedures is called management.

Member function
A function declared in the scope of a class to operate on data members.

Memory Management
This is the module of Operating System that manages memory in a system by deciding when to allocate and de-allocate memory to a process and how much to allocate.

Memory manager
A module of operating system responsible for the coordination of different types of computer memories by tracking which memory is available, which is to be allocated or de-allocated and how to move data among them.

Multiple inheritance
The concept of using more than one immediate base classes for a derived class is called multiple inheritance.

Multiprocessing
The simultaneous execution of two or more computer programs or sequences of instructions on multiple processors.

Multi-programming
A mode of operation in which two or more programs are executed in an interleaved manner by a single CPU.

Multitasking
Multitasking is the logical extension of multiprogramming in which multiple tasks are executed by a single CPU.

Multitasking operating system
Those operating systems which perform multiple tasks simultaneous on a single processor in such a way that creates the impression of parallel executions.

Multithreading
Multithreading is the ability of operating system that have multiple threads active in memory at the same time and the processor is switched quickly among them for processing.

Multi-user
Multi-user is the ability of an operating system that makes it possible for several users to login at the same time on a single computer.

Nested if-else
The use of one if statement in the body of another if statement is called nested if structure.

Nested loops
The use of a loop inside the body of another loop is called nested loop.

Non parameterized constructor
It is [illegible] those object declaration is done without specifying arguments.

Object
A variable of type class is called object. Object has attributes and functions which are combined together in a class construct.

Object oriented language
A programming language that allows the user to express a program in terms of objects and messages between those objects. Examples include C++, and Java etc.

ofstream
It is a file stream used for output that is defined in the stream "fstream".

One dimentional array
A one-dimensional array (or single dimension array) is a type of linear array which can be represented by a single subscript.

Operating System
System software that controls the execution of programs, and that provides services such as resource allocation, scheduling, input/output control, and data management.

Operational feasibility
In this type of feasibility it is determined that whether the proposed system will be used effectively after it has been developed or not.

Operator
The symbols used to perform operations over the operands. Examples are +, *, and &.

Output stream
are used to hold output for a particular data consumer, such as a monitor, a file, or a printer.

Overloading
Having more than one functions with the same name in the same scope or having more than one operator with the same name in the same scope.

Parallel processing
The ability to process "chunks" or blocks of a program simultaneously. To do this, the computer has to have several CPU units and the software to coordinate them.

Parameter
A value or reference passed to a function, or program that serves as input or controls actions.

Parameter
A variable declared in a function prototype or declarator for representing formal argument.

Peripheral devices
Any device which is attached to a computer system or LAN. Devices such as printers, disks and tape drives which are often attached to computers or LANs.

Pointer
A pointer is a variable that holds the address of another variable.

Polymorphism
Polymorphism is the ability to use an operator or function in different ways. Polymorphism gives different meanings or functions to the operators or functions.

Precedence of operator
means that which operand will be evaluated first and which one will be evaluated later if the expression is a complex one.

Preprocessor
The part of a C++ implementation that removes comments, performs macro substitution and #includes.

Private
An access control keyword used in classes. A member defined as private can only be accessed within the class and by the friends of the class.

Private access specifier
It tells the compiler that the members defined in a class by preceding this specifier are accessible only within the class and its friend function.

Private member
A member defined in the scope of private access specifier and accessible only from its own class is called private member.

Process
A job under execution is called process.

Process Control Block
A process control block (PCB) is a data structure that contains information related to a running process.

Process States
A status of a process at a particular time in which it may be is called process state. The process can be in one of the three states: Ready, Running or Suspended.

Processor Management
This relates to the Operating System's activity of managing the processor in the system, i.e. allocating and de-allocating of the processor to the process.

Programmer
Programmer is a technical person that writes computer programs in computer programming languages to develop software.

Programming language
A language used to express computer programs.

Project manager
is a professional in the field of project management responsible for planning, execution, and closing of any project.

Pseudo code
Pseudo code is an outline of a program, written in a form that can easily be converted into real programming statements.

Public
Access control keyword that is used to define public member (s) of a class is called public.

Public member
A member of a class defined by using the keyword public which is then accessible throughout the program is called public member.

Queue
A list formed by users programs (jobs) in a system waiting for their turns to process by the processor.

RAM
Random Access Memory: The memory that is available on a computer for storing data and programs that are currently being processed.

Ready state
A process state in which all resources except the processor are available.

Real Time Operating Systems
A Real Time Operating System (RTOS) is one in which the response to an event takes place in real time (in other words, as and when it is required). A typical example of a RTOS is the operating system used for flight control.

Register
A fast memory like data storage element which is part of the CPU or other control unit.

Relational operators
Are used for the comparison between two expressions. These operators are (==,=, >, >=, <, <=

Requirement validation
Concern with examining the requirements to certify that they meet the intentions of the stakeholders or not.

Requirements analysis
The process of studying user needs to arrive at a definition of a system, hardware, or software requirements.

Requirements phase
The phase of software development life cycle during which the requirements, such as functional and performance capabilities for a software product, are defined and documented.

ROM
Read Only Memory: Stored permanent systems instructions which are never changed; ROM is generally installed by the manufacturer as part of the system.

S

Schedule feasibility
It means that a project can be implemented in an acceptable time frame or not.

Scope
A region within a C++ program in which a C++ object or variable or function is visible is called its scope. It is usually delimited by curly braces { ... }.

Scope of a variable
By the scope of a variable we mean that if a variable is defined somewhere in a program then in which parts of the same program this variable can be accessed.

Secondary storage
Online peripheral data storage where data is permanently stored. Examples are magnetic disk (hard disk), DVD disks and CD disks etc.

Secondary storage management
The function of operating system to manage secondary storage devices and handle proper flow of data among primary and secondary storage devices is called secondary storage management.

Short
It is an integer type holding 2 bytes of memory and used to define integer variables.

Size of array
The number of elements that can be stored in an array is called the size or length of that array.

Software maintenance
Maintenance to a software system includes correcting software errors, adapting software to a new environment, or making enhancements to software i.e. adaptive maintenance, corrective maintenance, and perfective maintenance.

Source code
The human readable version of the list of instructions (program) that cause a computer to perform a task.

Stakeholders
Those entities which are either within the organization or outside of the organization that sponsor a project, or have an interest or have the intention to get it after its successful completion, or may have a positive or negative influence in the project completion are called stakeholders.

Standard stream
Standard streams in C++ consist of four predefined standard stream objects. These are cin, cout, cerr and clog.

Statement
In C++, a line of code ending with a semicolon symbol (;) is called statement.

Statement terminator
Each C++ statement is terminated by a symbol named semicolon (;) which is called statement terminator.

Static variable
A variable defined by using the keyword static is called static variable.

String
A sequence of character is called string.

String stream
A string stream is a stream which reads input from or writes output to an associated string.

switch-statement
Switch statement is a better alternative of the nested if-else structure which selects among many alternatives based on an integer value specified in switch expression.

Syst
It can be defined as a set of interrelated components having a clearly defined boundary that work together to achieve a common set of objectives.

System analysis
A systematic investigation of a real or planned system to determine the functions of the system and how they relate to each other and to any other system.

System design
A process of defining the hardware and software architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.

System Development Life Cycle (SDLC)
SDLC is a step wise process of creating computer systems. It is also known as information system development or application development.

System support
Keeping a system in its proper working condition is called maintenance.

T

Technical feasibility
Refers to the technical resources needed for the development of the proposed software system.

Ternary operator
An operator taking three operands, such as ? :).

Tester
Also called software tester who is a computer programmer having specialty in testing the computer programs using different testing techniques.

Testing
The execution of a program to find its errors is called testing.

Text file
Text files are those files that store data in text format which is readable by human and printable by the printers to make its hard copy.

Thread
A thread is a light weight process. It is the smallest unit of CPU utilization and is also the path of execution within a process.

Time sharing
Time sharing operating systems are those operating systems which slice processor (CPU) time into small fixed time slices and assign the processes to the CPU for that time slice to be executed.

Traversing of array
Accessing the elements of an array only once for the purpose of an operation is called traversing of array.

Two dimensional array
An array having more than one subscripts, i.e. X[i] [j]..[n], is called multidimensional array.

Unary operator
An operator taking one operand, such as, NOT operator (!), increment operator (++) or decrement (--) is called unary operator.

UNIX
Unix is a multitasking and multi-user operating system that was developed at Bell Labs in 1969.

User-defined type function
A function defined by the users for their own task is called user-defined function.

Variable
A name, label, identifier whose value may be changed many times during processing.

Virtual Memory
Virtual Memory refers to the concept whereby a process with a larger size than available memory can be loaded and executed by loading the process in parts.

Void
A keyword used to indicate an absence of information.

Waiting state
A waiting state is the blocked state of a process in which the process either waiting for its turn to be assigned to the CPU for execution or an external event to occur.

While-statement
An iterative statement that presents its condition "at the top" and is usually used for situations in which we do not know the exact number of iterations in advance is called while-statement or loop.

White-box testing
It is a procedure of testing software that tests internal structures or workings of an application.

Window
Windows operating systems are the most commonly used operating systems that are based on Graphical User Interfaces (GUI).

Index

Annexure 1

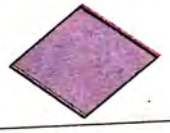
List of Figures

Annexure 2

List of Tables

Annexure 3

List of Symbols

Symbol	Name	Use
	Oval	Start/ End of Program
	Flow Line	Direction of Flow
	Parallelogram	Input or Output
	Rectangle	Processing
	Diamond	Decision Making

Annexure 4

List of Commands

Command	Action
CD	changes the current directory
COPY	copies a file
DEL	deletes a file
DIR	lists directory contents
EDIT	starts an editor to create or edit plain text files
FORMAT	formats a disk to accept DOS files
HELP	displays information about a command
MKDIR	creates a new directory
RD	removes a directory
REN	renames a file
TYPE	displays contents of a file on the screen

Annexure 5

List of Built-in Functions

Function	Header File	Purpose
getch()	*conio.h*	Used to get the character input from the keyboard
getche()	*conio.h*	Used to get character input form the keyboard and echo it on the screen
gets()	*stdio.h*	Stands for get string and is used to get a string input
puts()	*stdio.h*	Used to display the string
getchar()	*stdio.h*	Stands for get character it is used to input as single character from the keyboard. This function receives no argument.
putchar()	*stdio.h*	Standard output function used to display the character, on the output screen, received as an argument.
strcat()	*string.h*	Stands for String concatenation. It combines two strings into a single string.
copy()	*string.h*	This function copy one string into another string.
strcpy()	*string.h*	It is also used to copy one string into another string
find()	*string.h*	To find a particular word or a character in a string
length()	*string.h*	Used to find the length of a string.
swap()	*string.h*	To swap the values of two strings.
sin()	*math.h*	Used to find the sin value of an angle.
cos()	*math.h*	Used to find the cos value of an angle.
tan()	*math.h*	Used to ding the tangent value of an anbgle.
cot()	*math.h*	Used to find the cotangent value of an angle.
abs()	*math.h*	Used to find the absolute value of a variable or expression.
pow()	*math.h*	Used to find the power of a number. It takes two arguments first shows the number and second shows the power to which the number is raised.
sqrt()	*math.h*	Used to find the square root of an number received as an argument.
open()	*fstream.h*	Used to open a file.
close()	*fstream.h*	Used to close a file.
bof()	*fstream.h*	used to set the pointers to the beginning of a file
eof()	*fstream.h*	used to set the pointers at the end of a file

nnexure 6

Complete list of ASCII Codes

ASCII code	symbol	Description
0	NULL	(Null character)
1	SOH	(Start of Header)
2	STX	(Start of Text)
3	ETX	(End of Text)
4	EOT	(End of Transmission)
5	ENQ	(Enquiry)
6	ACK	(Acknowledgement)
7	BEL	(Bell)
8	BS	(Backspace)
9	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowledgement)
22	SYN	(Synchronous idle)
23	ETB	(End of transmission block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
32		(space)
33	!	(exclamation mark)
34	"	(Quotation mark)
35	#	(Number sign)
36	$	(Dollar sign)
37	%	(Percent sign)
38	&	(Ampersand)
39	'	(Apostrophe)
40	(	(round brackets or parentheses)
41	)	(round brackets or parentheses)
42	*	(Asterisk)

ASCII code	symbol	Description
43	+	(Plus sign)
44	,	(Comma)
45	-	(Hyphen)
46	.	(Full stop , dot)
47	/	(Slash)
48	0	(number zero)
49	1	(number one)
50	2	(number two)
51	3	(number three)
52	4	(number four)
53	5	(number five)
54	6	(number six)
55	7	(number seven)
56	8	(number eight)
57	9	(number nine)
58	:	(Colon)
59	;	(Semicolon)
60	<	(Less-than sign)
61	=	(Equals sign)
62	>	(Greater-than sign ; Inequality)
63	?	(Question mark)
64	@	(At sign)
65	A	(Capital A)
66	B	(Capital B)
67	C	(Capital C)
68	D	(Capital D)
69	E	(Capital E)
70	F	(Capital F)
71	G	(Capital G)
72	H	(Capital H)
73	I	(Capital I)
74	J	(Capital J)
75	K	(Capital K)
76	L	(Capital L)
77	M	(Capital M)
78	N	(Capital N)
79	O	(Capital O)
80	P	(Capital P)
81	Q	(Capital Q)
82	R	(Capital R)
83	S	(Capital S)
84	T	(Capital T)
85	U	(Capital U)
86	V	(Capital V)
87	W	(Capital W)
88	X	(Capital X)
89	Y	(Capital Y)
90	Z	(Capital Z)
91	[	(square brackets or box brackets)

ASCII code	symbol	Description
92	\	(Backslash)
93	]	(square brackets or box brackets)
94	^	(Caret or circumflex accent)
95	_	(underscore, under strike, underbar or low line)
96	`	(Grave accent)
97	a	(Lowercase a)
98	b	(Lowercase b)
99	c	(Lowercase c)
100	d	(Lowercase d)
101	e	(Lowercase e)
102	f	(Lowercase f)
103	g	(Lowercase g)
104	h	(Lowercase h)
105	i	(Lowercase i)
106	j	(Lowercase j)
107	k	(Lowercase k)
108	l	(Lowercase l)
109	m	(Lowercase m)
110	n	(Lowercase n)
111	o	(Lowercase o)
112	p	(Lowercase p)
113	q	(Lowercase q)
114	r	(Lowercase r)
115	s	(Lowercase s)
116	t	(Lowercase t)
117	u	(Lowercase u)
118	v	(Lowercase v)
119	w	(Lowercase w)
120	x	(Lowercase x)
121	y	(Lowercase y)
122	z	(Lowercase z)
123	{	(curly brackets or braces)
124	|	(vertical-bar, vbar, vertical line or vertical slash)
125	}	(curly brackets or braces)
126	~	(Tilde ; swung dash)
127	DEL	(Delete)
128	Ç	(Majuscule C-cedilla)
129	ü	(letter "u" with umlaut or diaeresis ; "u-umlaut")
130	é	(letter "e" with acute accent or "e-acute")
131	â	(letter "a" with circumflex accent or "a-circumflex")
132	ä	(letter "a" with umlaut or diaeresis ; "a-umlaut")
133	à	(letter "a" with grave accent)
134	å	(letter "a" with a ring)
135	ç	(Minuscule c-cedilla)
136	ê	(letter "e" with circumflex accent or "e-circumflex")
137	ë	(letter "e" with umlaut or diaeresis ; "e-umlaut")
138	è	(letter "e" with grave accent)
139	ï	(letter "i" with umlaut or diaeresis ; "i-umlaut")
140	î	(letter "i" with circumflex accent or "i-circumflex")

ASCII code	symbol	Description
141	ì	(letter "i" with grave accent)
142	Ä	(letter "A" with umlaut or diaeresis ; "A-umlaut")
143	Å	(letter "A" with a ring)
144	É	(Capital letter "E" with acute accent or "E-acute")
145	æ	(Latin diphthong "ae")
146	Æ	(Latin diphthong "AE")
147	ô	(letter "o" with circumflex accent or "o-circumflex")
148	ö	(letter "o" with umlaut or diaeresis ; "o-umlaut")
149	ò	(letter "o" with grave accent)
150	û	(letter "u" with circumflex accent or "u-circumflex")
151	ù	(letter "u" with grave accent)
152	ÿ	(letter "y" with diaeresis)
153	Ö	(letter "O" with umlaut or diaeresis ; "O-umlaut")
154	Ü	(letter "U" with umlaut or diaeresis ; "U-umlaut")
155	ø	(slashed zero or empty set)
156	£	(Pound sign ; symboi for the pound sterling)
157	Ø	(slashed zero or empty set)
158	×	(multiplication sign)
159	ƒ	(function sign ; f with hook sign ; florin sign)
160	á	(letter "a" with acute accent or "a-acute")
161	í	(letter "i" with acute accent or "i-acute")
162	ó	(letter "o" with acute accent or "o-acute")
163	ú	(letter "u" with acute accent or "u-acute")
164	ñ	(letter "n" with tilde ; enye)
165	Ñ	(letter "N" with tilde : enye)
166	ª	(feminine ordinal indicator)
167	º	(masculine ordinal indicator)
168	¿	(Inverted question marks)
169	®	(Registered trademark symbol)
170	¬	(Logical negation symbol)
171	½	(One half)
172	¼	(Quarter or one fourth)
173	¡	(Inverted exclamation marks)
174	«	(Guillemets or angle quotes)
175	»	(Guillemets or angle quotes)
176	▒	
177	▓	
178	▓	
179	┃	(Box drawing character)
180	┤	(Box drawing character)
181	Á	(Capital letter "A" with acute accent or "A-acute")
182	Â	(letter "A" with circumflex accent or "A-circumflex")
183	À	(letter "A" with grave accent)
184	©	(Copyright symbol)
185	╣	(Box drawing character)
186	┃	(Box drawing character)
187	┓	(Box drawing character)
188	┛	(Box drawing character)
189	¢	(Cent symbol)
190	¥	(YEN and YUAN sign)
191	┐	(Box drawing character)
192	└	(Box drawing character)
193	┴	(Box drawing character)
194	┬	(Box drawing character)
195	├	(Box drawing character)
196	─	(Box drawing character)
197	┼	(Box drawing character)
198	ã	(letter "a" with tilde or "a-tilde")

ASCII code	symbol	Description
199	Ã	(letter "A" with tilde or "A-tilde")
200	╚	(Box drawing character)
201	╔	(Box drawing character)
202	╩	(Box drawing character)
203	╦	(Box drawing character)
204	╠	(Box drawing character)
205	═	(Box drawing character)
206	╬	(Box drawing character)
207	¤	(generic currency sign)
208	ð	(lowercase "eth")
209	Ð	(Capital letter "Eth")
210	Ê	(letter "E" with circumflex accent or "E-circumflex")
211	Ë	(letter "E" with umlaut or diaeresis ; "E-umlaut")
212	È	(letter "E" with grave accent)
213	ı	(lowercase dot less i)
214	Í	(Capital letter "I" with acute accent or "I-acute")
215	Î	(letter "I" with circumflex accent or "I-circumflex")
216	Ï	(letter "I" with umlaut or diaeresis ; "I-umlaut")
217	┘	(Box drawing character)
218	┌	(Box drawing character)
219	█	(Block)
220	▄	
221	¦	(vertical broken bar)
222	Ì	(letter "I" with grave accent)
223	▀	
224	Ó	(Capital letter "O" with acute accent or "O-acute")
225	ß	(letter "Eszett" ; "scharfes S" or "sharp S")
226	Ô	(letter "O" with circumflex accent or "O-circumflex")
227	Ò	(letter "O" with grave accent)
228	õ	(letter "o" with tilde or "o-tilde")
229	Õ	(letter "O" with tilde or "O-tilde")
230	µ	(Lowercase letter "Mu" ; micro sign or micron)
231	þ	(capital letter "Thorn")
232	Þ	(lowercase letter "thorn")
233	Ú	(Capital letter "U" with acute accent or "U-acute")
234	Û	(letter "U" with circumflex accent or "U-circumflex")
235	Ù	(letter "U" with grave accent)
236	ý	(letter "y" with acute accent)
237	Ý	(Capital letter "Y" with acute accent)
238	¯	(macron symbol)
239	´	(Acute accent)
240	¬	(Hyphen)
241	±	(Plus-minus sign)
242	_	(underline or underscore)
243	¾	(three quarters)
244	¶	(paragraph sign or pilcrow)
245	§	(Section sign)
246	÷	(The division sign ; Obelus)
247	¸	(cedilla)
248	°	(degree symbol)
249	¨	(Diaeresis)
250	•	(Interpunct or space dot)
251	¹	(superscript one)
252	³	(cube or superscript three)
253	²	(Square or superscript two)
254	■	(black square)
255	nbsp	(non-breaking space or no-break space)

About Authors

1. **RAHMAN ALI** earned M.Phil (MS) degree in Computer Science from the Department of Computer Science, University of Peshawar in 2009. He earned his Master (M.Sc) degree in Computer Science from Hazara University, Mansehra in 2005. He did his B.Sc and F.Sc from Govt. Post Graduate lehanzeb College, Saidu Sharif, Swat in 2002 and 1999 respectively. He did SSC from Govt. High School Kishawara, Swat in 1997. He also holds bachelor degree in Education (B.Ed), earned from University of Peshawar in 2006.

He served Govt. Degree College, Kabal, Swat as a lecturer in Computer Science selected from September 2007 to August 2009. He also served Institute of IT, University of Science & Technology, Bannu as a lecturer in Computer Science from August 2009 to December 2009.

Rahman Ali is serving is a Lecturer in Computer Science at Quaid-e-Azam College of Commerce, University of Peshawar since December 2009.

Ali has his MS major in Artificial Intelligence (AI) and worked in Natural Language Processing (NLP). He has worked in anaphora, ambiguity and ellipsis resolution, Part-of- Speech (POS) tagging, natural language interfaces to databases, corpus linguistics, and machine translation and transliteration. He has published his research work in well reputed journals and conferences.

Mohammad Khalid is Head of Computer Science Department at OPF Boys College, Islamabad. Throughout his educational career he has got first-class first division. He did his M.Sc in Computer Science from the University of Peshawar in 1996 and earned his B.Ed degree in Science from Institute of Education and Research, University of Peshawar in 1997. He has been instructor and teacher in Computer Science for the last seventeen years. He has vast teaching experience from Secondary to Masters Level. He has attended many national and International level seminars and meetings for the promotion of teaching Computer Science at different levels.

He has been member of the National Curriculum development team sicne 2007. He has contributed a great deal in writing the Curriculum of Computer Education from Grades VI-VIII and Curriculum of Computer Science from Grades IX-XII.

As an author he has written a textbook of Computer Science for grade IX for KPK Textbook Board and a textbook of Computer Education for grade VII for National Books Foundation.

Keeping in view his experience and expertise in the subject, this book will prove to be an asset both for the students and the teachers.

مومن تو آپس میں بھائی بھائی ہیں۔
پس اپنے دو بھائیوں میں صلح کرا دیا کرو۔
اور اللہ سے ڈرتے رہو۔ تاکہ تم پر رحم کیا جائے۔
(سورۃ الحجرات: ۱۰)

مومن تو آپس میں بھائی بھائی ہیں۔
پس اپنے دو بھائیوں میں صلح کرا دیا کرو۔
اور اللہ سے ڈرتے رہو تاکہ تم پر رحم کیا جائے۔

(سورۃ الحجرات: ۱۰)

قومی ترانہ

پاک سَر زمین شاد باد　　کشورِ حَسین شاد باد
تو نشانِ عَزم عالی شان　　ارضِ پاکستان
مرکزِ یقین شاد باد

پاک سَر زمین کا نظام　　قُوت اُخوت عوام
قوم ، ملک ، سلطنت　　پائندہ ، تابِندہ باد
شاد بَاد منزلِ مُراد

پرچمِ سِتارہ و ہلال　　رَہبرِ ترقی و کمال
ترجُمان ماضی شانِ حال　　جانِ اِستقبال
سایۂ خدائے ذُوالجلال

LEADING BOOKS PUBLISHER
JAMRUD ROAD, PESHAWAR

Printer	Price	Quantity	Code No.
Creative Printers & Publishers	184.00	8,962	PSP/CTV-44-228-1819(O)