

File Organization
2nd Year
Faculty of Computers & Informatics

Q1- Answer:

Byte Offset : The distance, measured in bytes, from the beginning of the file. The first byte in the file has an offset of 0, the second byte has an offset 1, and so on.

Indexing : An index is a tool for finding records in a file. It consists of a key field on which the index is searched and a reference field that tells where to find the data file record associated with a particular key.

Primary Key : is a minimal number of fields that uniquely identify each record. It is used as the primary method for accessing the records.

File Manager : The file manager is a piece of software in the operating system, and consists of several layered procedure, with the upper layer dealing with logical aspects of file manager and lower layers dealing with the physical aspects of file manager. It also allocate the I/O buffers.

Operator Overloading: Mechanism provided in C++ which allow the programmer to use operators defined with basic data types to be defined and used with user-defined data types.

Q2- Answer:

Data Hierarchy: All data items processed by a computer are reduced to a combination of zeros and ones. Such a data item is called a bit. Digits, letters, and special symbols are referred to as characters, every character is represented as a pattern of eight 1s and 0s (called bytes) A field is a group of characters (bytes) that convey meaning. A record is a group of related fields. A file is a group of related records.

The ways that are used to separate one record from the next:

- 1- Make a record a predictable number of bytes (fixed length record)
- 2- Make a record a predictable numbers of fields.
- 3- Begin each record with a length indicator (variable length record)

4- use an index to keep track of addresses

5- place a delimiter at the end of each record

Q3- Answer:

Two ways to combat fragmentation:

Coalescence: If two deleted available records are physically adjacent, they can be combined to form a single, larger available record space.

Compaction: A way of getting rid of all external fragmentation by sliding all the records together so there is no space lost between them.

Placement strategies such as first fit, best fit, and worst fit are helping methods to combat fragmentation.

Q4- Answer:

IOS (istream (ifstream, iostream(fstream)), ostream(iostream, ofstream))

Q5- Answer:

```
const int maxbuffersize = 200;

int WritePerson (ostream &stream, Person &p){

char buffer[maxbuffersize];

strcat(buffer, p.last); strcat(buffer, "|");

strcat(buffer, p.first); strcat(buffer, "|");

strcat(buffer, p.city); strcat(buffer, "|");

short length = strlen(buffer);

stream.write(&length, sizeof(length));

stream.write(&buffer, length);

}
```

```
Int ReadVariablePerson (istream &stream, Person &p){  
    short length;  
    stream.read(&length, sizeof(length));  
    char *buffer = new char[length+1];  
    stream.read(&buffer, length);  
    buffer[length] = 0;  
    istream strbuff(buffer);  
    strbuff >> p;  
    return 1;  
}
```

Q6- Answer:

- a- false
- b- false
- c- true
- d- false
- e- false
- f- false
- g- true
- h- false

Q7- Answer:

- a- the program terminates normally

b- Main Memory.....Device (file, keyboard, or screen)

c- it is full or forced to do this

d- fstream iostream

e- Offset (long int) seek direction (ios::beg, ios::end, ios::curr)

f- Basic data types, strings, and pointer value

g- for determining if the file open operation succeeded

h- Relative Record Number

i- tab, space

j- << (the stream extraction), >> (the stream insertion) read and write functions

k- ctrl-z ctrl-d

l- record count, record size and file organization method

Q8- Answer:

a- FILE *fd = fopen ("karam.txt", "r"); int fd = open ("karam.txt", O_WRONLY);
fstream fd ("karam.txt", ios::in)

b- fwrite (&ch, 1, 1, stdout), where ch is a character variable

c- cout << static_cast<void *> st

d- cout << *floatptr

e- cin.get(c) and c = cin.get()

f- cin.read(nine, 40)

g- cin.get(name, 20, '!') and cin.getline(name, 20, '!')

Q9- Answer:

a- (iii) none of eofbit, failbit or badbit is set

b- (iii) either the badbit is set, the failbit is set, or both are set

c- (iii) either the badbit is set, the failbit is set, or both are set

d- (i) input

e- (i) Unformatted

f- (ii) No

g- (ii) A pointer value

h- (i) Logical files

i- (ii) Fixed-length record

j- (iii) The C language level

k- (i) Open Operation

l- (ii) Writing data into files from memory

m- (ii) Ensure that buffers are flushed

n - (ii) Indexed, entry-sequenced data file