Page No.
Date

SPCC OB for UT 1

1) What is Declaration Statement? Give Example.

→ A declaration is a statement describing an identifier, such as the name of a variable or a function.
Declarations are important because they inform the compiler or interpreter what they identifying word means, and how the identified thing should be used.
A declaration may be optional or required depending on the programing language.

It is used for Reserving memory for variable. Amount ~~and~~ of or world of ~~variable~~ memory reserving and that value will ~~initialize~~ have to initialize in memory that is called declaration statement

DS = Declare storage → This is for reserving memory
DC = Declare constant → variable ko constant values initialize karne ke liye

Eg Amount of memory ko jab memory me reserve karna hota he us variable ko kisi value se initialize karna hota he tab use declaration statement bolte he

Example

A DS 1 — One ~~one~~ world of memory reserved for A
A DC '5' — 5 value ko A se initialize karna he using DC.
&
B DC '5' — 5 constant value initialize by

Page No.
Date

1) List out the various system software.

→

System Software:

System software is the s/w. which is required to run the hardware parts of computer and other application software

There are two types of system s/w:

1) Standard System software
2) Operating system software:

1) Standard System software:

It contains assembler, processor, linker, compiler, debugger, loader, editors etc.

2) Operating System software:

It includes all operating system software which can be used to create interface between hardware and user application.

For example: Windows, DOS, UNIX, Linux etc

i) Assembler.
ii) Macro processor
iii) Loader
iv) Linker
v) Compiler
vi) interpreter
vii) Device driver
viii) Editor.
ix) - Word processor. - Ms. editor
- Line editor ex. teliprinter
- ~~Screen~~ editor ex. set editor. in unix
- Stream editor → Notepad
- Structure editor - C, java, python
x) Debugger.

3) Define Application software :

-) - Application program is also called as application software
- Application software is mainly developed for end user to perform specific operations that he want to do. i.e. it is developed for some special purpose use.

e.g. word processor, spreadsheets, email clients, web browser.

Features of Application software
i) Easy to design
ii) Generally written in high-level language.
iii) More interactive
iv) Slow in speed
v) Close to the user
vi) Easy to manipulate and use.
vii) Bigger in size and required large storage space.
viii) Easy to understand.

OR

A set of programs that work together to solve particular problems of user through computer is called application software.

5) ... example.

4. What are advantages of Assembly language.

Advantages:

- Assembly language program increase readability with the help of mnemonic operation code.
- It is memory efficient
- It is not required to keep track of memory locations
- Assembly language is faster in speed.
- Easy to make insertion and deletions
- Assembly language is hardware oriented.
- Assembly language is useful in embedded system as it have less resources, it helps in managing size and code.
- Requires less instruction to compute the same result
- It can access hardware driver and system code easily which can be difficult using high level code.

5) Explain Assemble directives with Example.

→

~~There are 3 assemble directives~~

~~1) ORIGIN~~

~~2) EQU~~

~~3) LTORG~~

An assembler directive guides the assembler to perform certain actions during the assembly of program

Som assembler directives are described in the following

1) START <constant>

- This directive indicates that the 1st word generated by assembler is stored in memory location specified by <constant>

Example : START 400

2) END [<operand speci>]

- This directives specifies the end of the source program.
- The optional <operand spec> indicates the address of the instruction where the execution of the program should begin.

Page No.
Date

6) Define Compiler and Interpreter:

→ Compiler:

Compiler is a system program that translate high level language into low level language

- Compiler takes the whole source code at a time and checks for errors. If source code is error free then compiler directly translates it into object code.

Example gcc compiler

source code → [Compiler] → object code

Interpreter:

Interpreter is also a translator like computer but interpreter takes one line at a time and translates it into object code. then go for the next line and interprets.
This process continues until the entire source code is interpreted.

Example: The best ex. of interpreter is Java Byte Code Interpreter.

source code → [Interpreter] → object code

Date

7) Explain the features of system software?

→ Computer manufacturers develop the system s/w as an integral part computer. The primary responsibility of this s/w is to create an interface between computer hardware they manufacture and the end use

System software generally includes following features.

1) High speed: System s/w must be as efficient as possible to provide an effective platform for higher-level s/w in the comp system

2) Hard to manipulate: It often requires the use of programming language, which is more difficult to use than a more intuitive user interface (UI)

3) Written in a low-level computer language: System software must be written in a comp language the central processing unit and other computer hardware can read

4) Close to the system: It connects directly to the hardware that enables the computer to run

5) Versatile: System s/w must communicate with both the specialized hardware it runs on and the higher-level application s/w that is usually

hardware-agnostic and often has no direct connection to the hardware it runs on. System software also must support other programs that depends on it as they evolve and change.

6) Define Loader and Linker:

→

Loader:

Loader performs the function of placing object code into main memory for execution purpose. To do this, loader translates object code into executable form.

object code → [Loader] → Executable ~~source~~ code

Linker:

Computer programs contain several modules that expand separate object files, each of which are compiled. The program refers to these-external-compiled object files using notations. The Linker merges these separate files into a single program resolving the notational references as it goes long

In other word linker is used to modify the addresses of notations according to new memory location.

object code → [Linker] → Executable code

Page No.
Date

9) State goal of system s/w

→

- System s/w is the s/w which is required to run the hardware parts of computer and other application. s/w.
- System s/w provides the base of infrastructure for various s/w to run smoothly
- System s/w handles the hardware, s/w and network together
- In other words system s/w can be defined as an interface betwn the hardware and user applications.
- The system s/w converts all human instructions written in application program into machine understandable formats.
- System s/w is responsible for smooth functioning of all hardware parts and their interoperability to execute tasks successfully.
- SS manages basic functions of computer like storing data in the memory, retriving files from the storage devices.
- The best ex of SS is
  Operating system (OS)
  Compilers,
  Interpreter
  Assembler etc.

10) Define System Software.

→

System Software:

System software or system application, is computer s/w. designed to provide platform to other s/w.

System s/w is a set of programs which are developed to operate, control and extend processing capabilities of computer itself.

11) Explain advance assembly directives.

→

i) ORIGIN.

Syntax:

ORIGIN <address speci>

- where <address speci> is an operand specification or it may be constant. ~~(LC) specified~~ The ORIGIN directive is used to set the location constant (LC) specified in the address specification.

Ex.

1. ORIGIN 400

It set the LC to 400 memory location.

ii) EQU

Syntax:

<symbol> EQU <address speci>

- This directive can associate the symbol with the <address speci>. The new symbol gets the same address as specified in <address speci>

Example: RESULT EQU TOTAL

Here the symbol RESULT gets the same address as TOTAL

iii) LTORG

- By default assembler allocates memory for literal at the end of source program. The LTORG statement is used to specify where literals should be placed during the program execution.