

(Ques)

- Software processes are the activities for designing, implementing and testing a software. It represents the overall development process of a software. The software process Model is an abstract representation of development process.

Cohesion

Coupling

- | | |
|---|--|
| ① Cohesion is the indication of relationship within a module. | ① Coupling is the indication of relationships between modules. |
| ② It is concept of Intra-module. | ② It is concept of Inter module. |
| ③ Increasing Cohesion is good for software. | ③ Increasing coupling is avoided for software. |
| ④ It represents the functional strength of a module. | ④ Coupling represents the independence among modules. |
| ⑤ Highly cohesive gives the best software. | ⑤ Whereas loosely coupling gives the best software. |

⑥

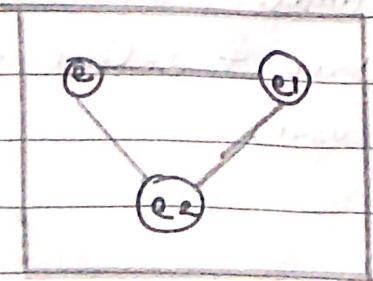


fig: cohesion.

⑥

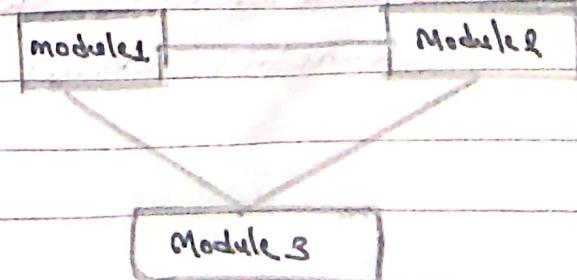


fig: coupling.

2.

→ functions are a separate blocks of codes which are programmed / designed to perform a specific kind of job. We can divide a whole program into basic blocks of codes known as function. Set of statement can be implemented within the function by {} . Readability, reusability & reaccessibility are beautiful features of function.

Syntax:

return type functionname(arguments);

// This is declaration.

functionname(arguments); // function call.

returntype = functionname(arguments){

// code to be executed

} // This is definition.

We can declare a function & multiple instruction can be implemented within a function.

Using a function we can implement our logic & we can use it multiple times.

To demonstrate we will declare & define a function which greets the user.

```
#include <stdio.h>
void greetuser( char name[20] );
int main()
{
    int i;
    char user[20];
    printf("Enter yourname: ");
    scanf("%s", user);
    greetuser(user);
    return 0;
}
void greetuser( char name[20] )
{
    printf("Namaskar, %s", name);
}
```

OUTPUT:

» Enter yourname: Gobinda

Namaskar, Gobinda.

Q3)

→ Pointers are the variables which holds the address of any other variable of any type. Use of pointer enhances the overall execution speed. It can handle array more efficiently. It reduces storage space. It help in building complex datastructure like linked list, stacks, queues, trees etc.

```
#include <stdio.h>
int fibo(int n);
int main() {
    int n=13, a=0, b=1, next, i;
    printf("fibonacci series upto 13th term are:\n");
    scanf("%d", &n);
    for( i=0; i<n; i++) {
        printf("\t%d\t", fibo(a));
        a++;
    }
    return 0;
}

int fibo (int n) {
    if (n==0 || n==1)
    {
        return n;
    }
    else {
        return (fibo(n-1) + fibo(n-2));
    }
}
```

Output:

fibonacci series upto 13th term are:

0 1 1 2 3 5 8 13 21 34 55 89 144

④

→ Loop means any task which has to be done or executed again repeatedly. It is the technique in any programming language to perform a same task multiple time or we can say to execute same line multiple time.

In C there are 3 kinds of loop-

they are:

① while loop

② do-while loop

③ for loop

Let us print 0 to 10 number using loop.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i;
```

```
    for(i=0; i<11; i++)
```

```
{
```

```
        printf("%d\n", i);
```

```
}
```

```
return 0;
```

```
}
```

Output

0 1 2 3 4 5 6 7 8 9 10

'break' keyword is useful for jumping out of the loop skipping code below it without caring the condition. To demonstrate 'break' keyword we'll write a program which print 1 to 5 numbers without & jump out of the loop without caring condition.

```
#include <stdio.h>
```

```
int main()
{
    int i=1;
    while (i<100){
        printf("%d\n", i);
        if (i>5){
            break;
        }
        i++;
    }
}
```

OUTPUT:

```
1 2 3 4 5
```

In above program while loop should iterate 100 time but due to break statement it jump out of loop when $i=5$. Hence 'break' helps to jump out of loop when needed.

Contd:- 'continue' statement is used when we have to skip some condition during the iteration through a condition in a loop.

To demonstrate a 'continue' statement we will

Write a program in C which prints odd number from 1 to 10.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i=1;
```

```
    for(i=1; i<=10; i++)
```

```
{
```

```
    if((i-1)*2 == 0)
```

```
{
```

```
        continue;
```

```
}
```

```
    printf("%d\n", i);
```

```
}
```

```
return 0;
```

OUTPUT:

0 1 3 5 7 9

In above code when the given condition inside the loop become true continue statement skips that iteration & jumps to next iteration.

(5)

classmate

Date _____

Page _____

→ DMA is dynamic Memory Allocation. It can be defined as a procedure in which the size of datastructure (like an array) can be changed during runtime. C provides some library function under <stdlib.h> header file to facilitate DMA in C.

These functions are:

- ① malloc
- ② calloc
- ③ free
- ④ realloc

```

#include < stdio.h>
#include <stdlib.h>

int main()
{
    int *dataloc;
    int n, i;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    dataloc = (int *) malloc(n * sizeof(int));
    for(i=0; i<n; i++)
    {
        printf("Enter number %d: ", i+1);
        scanf("%d", (dataloc+i));
    }

    int min = *dataloc;
    int max = *dataloc;

    for(i=0; i<n; i++)
    {
        if(max < *(dataloc+i))
        {
            max = *(dataloc+i);
        }
        if(min > *(dataloc+i))
        {
            min = *(dataloc+i);
        }
    }

    printf("\n\nmax=%d \n min=%d", max, min);
    return 0;
}

```

Output:

Enter number of elements: 5

Enter number1: 1

Enter number2: 3

Enter number3: 4

Enter number4: 7

Enter number5: 0

max = 7

min = 0

⑥

Differences between binary file & Textfile.

Text file

① It can store only plain text.

② Bits represents character.

③ It can be opened using any text editor.

④ extensions are: .txt, .rtf

Binary file

① can store different types of data (image, audio, video) in a single file.

② Bits represents custom data.

③ only the specified application can open this file.

④ can have any kind of extensions.

program:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
```

FILE *ptr;

char message[20];

```
ptr = fopen ("test.txt", "w");
```

```
fprintf (ptr, "Welcome to BCA", message);
```

```
fclose(ptr);
```

```
return 0;
```

}

7)

→ Structures are the user defined datatype in C in which we can store different variable of data of various datatypes. Each data object in a structure is a member or field.

Syntax:

```
struct struct-name {
```

```
datatype variable;
```

```
datatype variable;
```

```
:
```

```
:
```

```
} - var1, var2...;
```

→ A Union is an object similar to a structure except that all of its member starts at the same memory location in memory. A union variable can represent the value of only one of its member at a time.

Syntax:

```
Union Union-name {
```

```
datatype var1;
```

```
datatype var2;
```

```
} var1, var2...;
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct students {
```

```
    char name[20];
```

```
    int roll;
```

```
    char address[30];
```

```
    int marks;
```

```
}
```

```
int main()
```

```
    struct students s[35];
```

```
    for (int i=0; i<35; i++) {
```

```
        s[i].roll = i+1;
```

```
        printf("Enter data for student having roll no. %d", s[i].roll);
```

```
        printf("Enter name: ");
```

```
        gets(s[i].name);
```

```
        fflush(stdin);
```

```
        printf("Enter address: ");
```

```
        scanf("%s", &s[i].address);
```

```
        printf("Enter marks: ");
```

```
        scanf("%d", &s[i].marks);
```

```
}
```

```
printf("Students who obtained marks > 250 are:\n");
```

```
for (int i=0; i<35; i++) {
```

```
    if (s[i].marks > 250) {
```

```
        printf("Name: %s, Roll No. %d, Address: %s,\n",
```

```
               s[i].name, s[i].roll, s[i].address);
```

```
        printf("\n");
```

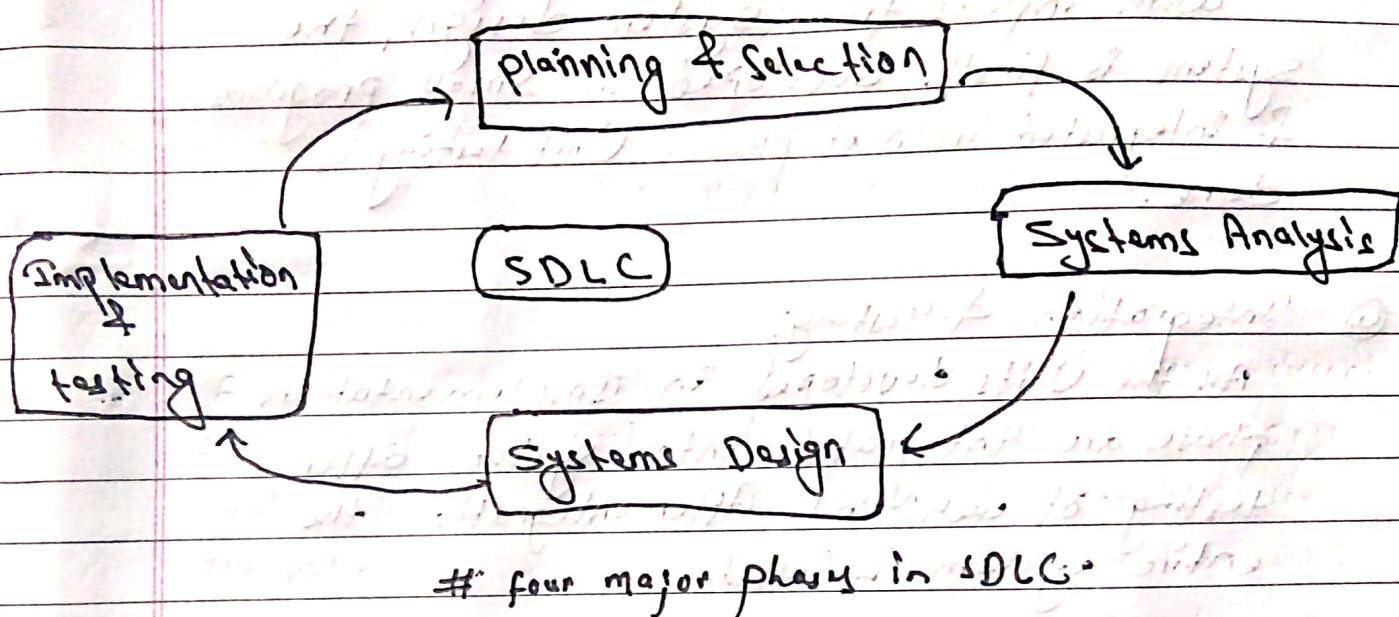
```
}
```

```
}
```

```
}
```

⑨

→ Software development life cycle (SDLC) is a process used by the software industry to develop design & test high quality softwares. The SDLC aims to produce high quality software that meets or exceeds customers expectations, reaches completion within time & cost estimate.



→ Waterfall model was first SDLC model to be used widely in software engineering to ensure the success of the project. In this model outcome of one phase acts as another next input for another phase sequentially.

phases in waterfall model are:-

① Requirement analysis:-

- All requirements of the system to be developed are captured & analyzed & documented in specification document.

② System Design:

- The requirement specification from 1st phase is studied & system design is prepared.
- It helps to specifying hardware & overall system architecture.

③ Implementation:

With inputs from system design, the system is firstly developed in small programs & integrated in next phase. Unit testing is done.

④ Integration & Testing:

All the units developed in implementation phase are integrated into system after testing of each unit. After integration the entire system is tested.

⑤ Deployment of system:

Once the functional testing is done the product is deployed to customer or market.

⑥ Maintenance:

If any issues come up in the client environment. These are fixed, another version is released.

Requirement Analysis

System design

Implementation

Testing

Deployment

Maintainance

fig: waterfall model.

Advantages

- simple easy to understand & use.
- easy to manage due to rigidity of project model.
- clearly defined stages.
- easy to arrange tasks
- phases are processed & completed one at a time.
- process & results are well documented.

Disadvantages

- High amount of risk & Uncertainty.
- Not a good model for complex & oop projects.
- poor model for large & ongoing projects.
- It is difficult to measure progress within stages.
- cannot accommodate changing requirement.