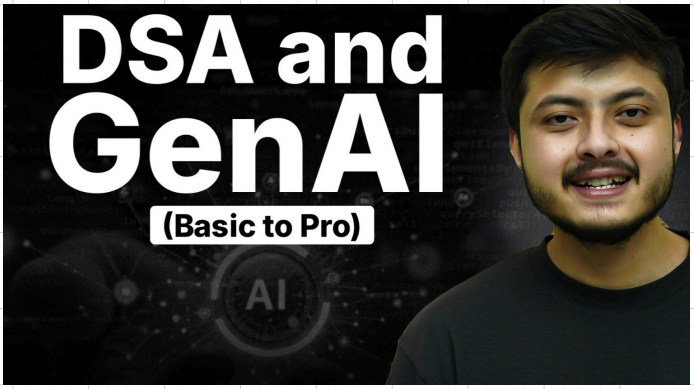


DSA - 6

Why Functions?

Problem:

```
int num1 = 3, num2 = 5, i = 1;
long long fact = 1;

while(i <= num1) {
   fact *= i;
   i++;
}
cout<<fact;

while(i <= num2) {
   fact *= i;
   i++;
}
cout<<fact;
```

Solution:

```
int factorial (int n) {

   int fact  = 1;
   int i = 1;
   while(i<=n) {
      fact *= i;
      i++;
   }
   return fact;
}
int main() {
   int num1 = 3, num2 = 5;
   cout<<factorial(num1);
   cout<<factorial(num2);
}
```

Instead of writing the same code again and again, write it once in a function and call it whenever you need it.

Function → DRY: Don't Repeat Yourself

What is Function?

A function is a reusable block of code designed to perform a specific task.

Syntax:

```
return-type function_name (parameter1, parameter2, ....) {

   // function body

   return value;
}
```

```
int square (int, int);          -> function declaration

int main(){
   int result = square(5);  -> function call
   cout<<result;
}
int square (int a) {
   return a*a;                  -> function definition
}
```

Real-Life Example: YouTube Home Screen

Each video has:

1. Thumbnail
2. Channel name
3. Title
4. View count
5. Upload time

```
displayVideo(thumbnail, channelName, title, views, uploadTime)
```

Function Overloading:

Multiple functions with the same name but different parameter lists.

How Compiler Chooses the Function?

This is called compile-time polymorphism

- Function name
- Number of arguments
- Data types of arguments

```
int sum(int a, int b) {
   return a+b;
}

int sum(int a, int b, int c) {
   return a+b+c;
}

float sum(float a, float b) {
   return a+b;
}
```

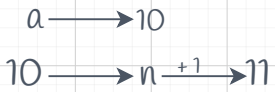
Pass By Value:

```
void increment (int n){
   n++;
}

int main() {
   int a = 10;
   increment(a);
   cout<<a<<endl;
}
```

Output: 10

A copy of the variable is passed to the function.



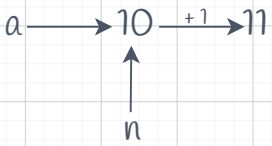
Pass By Reference:

```
void increment (int &n){
   n++;
}

int main() {
   int a = 10;
   increment(a);
   cout<<a<<endl;
}
```

Output: 11

Address of the variable is passed to the function.



Default Parameter:

```
void print(int num = 5) {
   cout<<num<<endl;
}

int main() {
   print();
}
```

A default parameter is a parameter that has a predefined value.

If no argument is passed during the function call, the default value is used.