STACK OPERATION IN C PROGRAMMING

```
#include<stdio.h>
#include<stdlib.h>
#define Size 4
int Top=-1, inp_array[Size];
void Push();
void Pop();
void show();
int main()
{
   int choice;

   while(1)
   {
      printf("\nOperations performed by Stack");
      printf("\n1.Push the element\n2.Pop the
element\n3.Show\n4.End");
      printf("\n\nEnter the choice:");
      scanf("%d",&choice);
      switch(choice)
      {
         case 1: Push();
               break;
         case 2: Pop();
               break;
         case 3: show();
               break;
         case 4: exit(0);

         default: printf("\nInvalid choice!!");
      }
   }
}
void Push()
{
   int x;

   if(Top==Size-1)
   {
      printf("\nOverflow!!");
   }
   else
   {
```

STACK OPERATION IN C++

```
#include <iostream>
using namespace std;
int stack[100], n=100, top=-1;
void push(int val) {
   if(top>=n-1)
   cout<<"Stack Overflow"<<endl;
   else {
      top++;
      stack[top]=val;
   }
}
void pop() {
   if(top<=-1)
   cout<<"Stack Underflow"<<endl;
   else {
      cout<<"The popped element is "<< stack[top]
<<endl;
      top--;
   }
}
void display() {
   if(top>=0) {
      cout<<"Stack elements are:";
      for(int i=top; i>=0; i--)
      cout<<stack[i]<<" ";
      cout<<endl;
   } else
   cout<<"Stack is empty";
}
int main() {
   int ch, val;
   cout<<"1) Push in stack"<<endl;
   cout<<"2) Pop from stack"<<endl;
   cout<<"3) Display stack"<<endl;
   cout<<"4) Exit"<<endl;
   do {
      cout<<"Enter choice: "<<endl;
      cin>>ch;
      switch(ch) {
         case 1: {
            cout<<"Enter value to be pushed:"<<endl;
            cin>>val;
```

```
      printf("\nEnter element to be inserted to the
stack:");
       scanf("%d",&x);
       Top=Top+1;
       inp_array[Top]=x;
   }
}
void Pop()
{
   if(Top==-1)
   {
      printf("\nUnderflow!!");
   }
   else
   {
      printf("\nPopped element:  %d",inp_array[Top]);
      Top=Top-1;
   }
}
void show()
{


   if(Top==-1)
   {
      printf("\nUnderflow!!");
   }
   else
   {
      printf("\nElements present in the stack: \n");
      for(int i=Top;i>=0;--i)
         printf("%d\n",inp_array[i]);
   }
}
```

OUTPUT

Operations performed by Stack
1.Push the element
2.Pop the element
3.Show
4.End
Enter the choice:1

```
      push(val);
      break;
    }
    case 2: {
      pop();
      break;
    }
    case 3: {
      display();
      break;
    }
    case 4: {
      cout<<"Exit"<<endl;
      break;
    }
    default: {
      cout<<"Invalid Choice"<<endl;
    }
   }
  }while(ch!=4);
  return 0;
}
```

OUTPUT

1) Push in stack
2) Pop from stack
3) Display stack
4) Exit
Enter choice: 1
Enter value to be pushed: 2

```
Enter element to be inserted to the stack:10
Operations performed by Stack
1.Push the element
2.Pop the element
3.Show
4.End
Enter the choice:3
Elements present in the stack:
10
Operations performed by Stack
1.Push the element
2.Pop the element
3.Show
4.End
Enter the choice:2
Popped element:  10
Operations performed by Stack
1.Push the element
2.Pop the element
3.Show
4.End
Enter the choice:3
Underflow!!
```

```
Enter choice: 1
Enter value to be pushed: 6
Enter choice: 1
Enter value to be pushed: 8
Enter choice: 1
Enter value to be pushed: 7
Enter choice: 2
The popped element is 7
Enter choice: 3
Stack elements are:8 6 2
Enter choice: 5
Invalid Choice
Enter choice: 4
Exit
```

C LANGUAGE

- C was developed by Dennis Ritchie between the year 1969 and 1973 at AT&T Bell Labs.
- C does no support polymorphism, encapsulation, and inheritance which means that C does not support object oriented programming.
- C is a subset of C++.
- C contains 32 keywords.
- For the development of code, C supports procedural programming.
- Data and functions are separated in C because it is a procedural programming language.
- C does not support information hiding.
- Built-in data types is supported in C.
- C is a function driven language because C is a procedural programming language.
- Function and operator overloading is not supported in C.
- Functions in C are not defined inside structures.
- Namespace features are not present inside the C.
- Header file used by C is stdio.h.
- Reference variables are not supported by C.
- Virtual and friend functions are not supported by C.
- C does not support inheritance.
- Instead of focusing on data, C focuses on method or process.

C++ LANGUAGE

- C++ was developed by Bjarne Stroustrup in 1979.
- C++ supports polymorphism, encapsulation, and inheritance because it is an object oriented programming language.
- C++ is a superset of C.
- C++ contains 63 keywords.
- C++ is known as hybrid language because C++ supports both procedural and object oriented programming paradigms.
- Data and functions are encapsulated together in form of an object in C++.
- Data is hidden by the Encapsulation to ensure that data structures and operators are used as intended.
- Built-in & user-defined data types is supported in C++.
- C++ is an object driven language because it is an object oriented programming.
- Function and operator overloading is supported by C++.
- Functions can be used inside a structure in C++.
- Namespace is used by C++, which avoid name collisions
- Header file used by C++ is iostream.h.
- Reference variables are supported by C++.
- Virtual and friend functions are supported by C++.
- C++ supports inheritance.

- C provides malloc() and calloc() functions for dynamic memory allocation, and free() for memory de-allocation.
- Direct support for exception handling is not supported by C.
- scanf() and printf() functions are used for input/output in C.
- C structures don't have access modifiers.

- C++ focuses on data instead of focusing on method or procedure.
- C++ provides new operator for memory allocation and delete operator for memory de-allocation.
- Exception handling is supported by C++.
- cin and cout are used for input/output in C++
- C ++ structures have access modifiers.