```
//java program to demonstrate stack operations
#include<iostream>
#include<string>

using namespace std;

class Stack {
 private:
  int top;
 int arr[5];

 public:
  Stack() {
   top = -1;
   for (int i = 0; i < 5; i++) {
    arr[i] = 0;
   }
  }

 bool isEmpty() {
  if (top == -1)
   return true;
  else
   return false;
 }
 bool isFull() {
  if (top == 4)
   return true;
  else
   return false;
 }

 void push(int val) {
  if (isFull()) {
   cout << "stack overflow" << endl;
  } else {
   top++; // 1
   arr[top] = val;
  }
 }

 int pop() {
  if (isEmpty()) {
   cout << "stack underflow" << endl;
   return 0;
  } else {
   int popValue = arr[top];
   arr[top] = 0;
   top--;
   return popValue;
  }
 }
```

```
  int count() {
    return (top + 1);
  }

  int peek(int pos) {
    if (isEmpty()) {
      cout << "stack underflow" << endl;
      return 0;
    } else {
      return arr[pos];
    }
  }

  void change(int pos, int val) {
    arr[pos] = val;
    cout << "value changed at location " << pos << endl;
  }

  void display() {
    cout << "All values in the Stack are " << endl;
    for (int i = 4; i >= 0; i--) {
      cout << arr[i] << endl;
    }
  }
};

int main() {
  Stack s1;
  int option, postion, value;

  do {
    cout << "What operation do you want to perform? Select Option number. Enter 0 to exit." << endl;
    cout << "1. Push()" << endl;
    cout << "2. Pop()" << endl;
    cout << "3. isEmpty()" << endl;
    cout << "4. isFull()" << endl;
    cout << "5. peek()" << endl;
    cout << "6. count()" << endl;
    cout << "7. change()" << endl;
    cout << "8. display()" << endl;
    cout << "9. Clear Screen" << endl << endl;

    cin >> option;
    switch (option) {
    case 0:
      break;
    case 1:
      cout << "Enter an item to push in the stack" << endl;
      cin >> value;
      s1.push(value);
      break;
    case 2:
```

```
    cout << "Pop Function Called - Poped Value: " << s1.pop() << endl;
    break;
   case 3:
    if (s1.isEmpty())
     cout << "Stack is Empty" << endl;
    else
     cout << "Stack is not Empty" << endl;
    break;
   case 4:
    if (s1.isFull())
     cout << "Stack is Full" << endl;
    else
     cout << "Stack is not Full" << endl;
    break;
   case 5:
    cout << "Enter position of item you want to peek: " << endl;
    cin >> postion;
    cout << "Peek Function Called - Value at position " << postion << " is " << s1.peek(postion) << endl;
    break;
   case 6:
    cout << "Count Function Called - Number of Items in the Stack are: " << s1.count() << endl;
    break;
   case 7:
    cout << "Change Function Called - " << endl;
    cout << "Enter position of item you want to change : ";
    cin >> postion;
    cout << endl;
    cout << "Enter value of item you want to change : ";
    cin >> value;
    s1.change(postion, value);
    break;
   case 8:
    cout << "Display Function Called - " << endl;
    s1.display();
    break;
   case 9:
    system("cls");
    break;
   default:
    cout << "Enter Proper Option number " << endl;
   }

  } while (option != 0);

  return 0;
}
```