2. Develop a stack class to hold a maximum of 10 integers with suitable methods. Develop a JAVA

main method to illustrate Stack operations

Save Filename as:StackMain.java

Solution:-

```
import java.util.Scanner;
class Stack
{
private int maxSize = 10;
private int top;
private int[] stackArray;
public Stack ()
{
stackArray = new int[maxSize];
top = -1;
}
public void
push (int value)
{
if (top == maxSize - 1)
{
System.out.println("Stack is full. Unable to
push " + value);
return;
}
stackArray[++top] = value;
}
public void
pop ()
{
if (top == -1)
{
System.out.println ("Stack is empty");
return;
}
System.out.println ("Popped " + stackArray[top--] + "from the stack");
}
public void
display ()
{
if (top == -1)
{
System.out.println ("Stack is empty");
return;
}
System.out.print ("Stack: ");
for (int i = 0; i <= top; i++)
```

```
{
2
System.out.print (stackArray[i] + " ");
}
System.out.println ();
}
}
public class StackMain
{
public static void main (String[] args)
{
Stack stack = new Stack ();
Scanner scanner = new Scanner (System.in);
while (true)
{
System.out.println ("Choose an option:");
System.out.println ("1) Push");
System.out.println ("2) Pop");
System.out.println ("3) Display");
System.out.println ("4) Exit");
int option = scanner.nextInt ();
switch (option)
{
case 1:
```

```
System.out.println ("Enter a number to push:");
int num = scanner.nextInt ();
stack.push (num);
break;
case 2:
stack.pop ();
break;
case 3:
stack.display ();
break;
case 4:
scanner.close ();
return;
default:
System.out.println("Invalid option.Please
choose again.");
}
}
}
}
```