

## Lab Activity 3

Name : Sarvagya Gupta

Sap ID : 500083195

Batch : B6

---

Title - Basic Java Programming

- 1) Write a program to accept three digits (i.e., 0 - 9) and print all its possible combinations. (For example, if the three digits are 1, 2, 3 then all possible combinations are: 123, 132, 321)

### Code

```
lab3_ques1.java > ...
1  import java.util.Scanner;
2
3  public class lab3_ques1{
    Run | Debug
4      public static void main(String[] args) {
5          System.out.println("Enter any 3 non-repeating digits");
6          Scanner sc= new Scanner(System.in);
7          int n1 = sc.nextInt();
8          int n2= sc.nextInt();
9          int n3= sc.nextInt();
10         if(n1==n2 || n2==n3 || n3==n1) {
11             System.out.println("You entered repeating digits.");
12         }
13         else{
14             String x=String.valueOf(n1);
15             String y=String.valueOf(n2);
16             String z=String.valueOf(n3);
17             System.out.println("All possible outcomes are: ");
18             System.out.println(x+y+z+" , "+y+z+x+" , "+z+x+y+" , "+z+y+x+" , "+x+z+y+" , "+y+x+z);
19         }
20     }
21 }
22
23 }
24
```

### Output

```
PS E:\codes\java\lab 3> & 'c:\Users\Sarvagya Gupta\.vscode\extensions\vscjava.vscod-jav
0.12.7-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\Sarvagya Gupta\AppData
a\jdk_ws\lab 3_af84e7f6\bin' 'lab3_ques1'
Enter any 3 non-repeating digits
1 4 2
All possible outcomes are:
142 , 421 , 214 , 241 , 124 , 412
PS E:\codes\java\lab 3> █
```

- 2) Write a Java Program to accept 10 numbers in an array and compute the square of each number. Print the sum of these numbers.

## Code

```
lab3_ques2.java > lab3_ques2
1  import java.util.Scanner;
2
3  public class lab3_ques2{
    Run | Debug
4      public static void main(String[] args){
5          Scanner sc= new Scanner(System.in);
6          System.out.println("Enter elements of array:");
7          int x=0;
8          int arr[]= new int[10];
9          int arr2[]= new int[10];
10         for(int i=0;i<10;i++){
11             arr[i]= sc.nextInt();
12         }
13         for(int i=0;i<10;i++){
14             arr2[i]= arr[i]*arr[i];
15         }
16         for(int i=0;i<10;i++){
17             x = x + arr[i];
18         }
19         System.out.println("The square of all numbers are: ");
20         for(int i=0;i<10;i++){
21             System.out.print(arr2[i]+" ");
22         }
23         System.out.println("");
24         System.out.println("The sum of all numbers is = "+ x);
25     }
26 }
```

## Output

```
PS E:\codes\java\lab 3> e.; cd e:\codes\java\lab 3 ; &
les\Eclipse Foundation\jdk-11.0.12.7-hotspot\bin\java.exe
5f010fd487cbc73b17a\redhat.java\jdt_ws\lab 3_af84e7f6\bin
Enter elements of array:
3 6 2 1 9 7 9 3 1 0
The square of all numbers are:
9 36 4 1 81 49 81 9 1 0
The sum of all numbers is = 41
PS E:\codes\java\lab 3>
```

- 3) Write a program to input a number of a month (1 - 12) and print its equivalent name of the month. ( e.g 1 to Jan, 2 to Feb. 12 to Dec.)

## Code

```
lab3 ques3.java > lab3 ques3 > main(String[])
1  import java.util.Scanner;
2
3  public class lab3 ques3{
    Run | Debug
4      public static void main(String[] args) {
5          System.out.println("Enter the number of the month: ");
6          Scanner sc = new Scanner(System.in);
7          int n1 = sc.nextInt();
8          String[] arr=new String[] {"JANUARY","FEBRUARY","MARCH","APRIL","MAY","JUNE","JULY","AUGUST","SEPTEMBER","OCTOBER",
9              "NOVEMBER","DECEMBER"};
10         if(n1>=0 && n1<=12) {
11             System.out.println("Respective month is : "+ arr[n1 - 1]);
12         }
13         else {
14             System.out.println("Invalid number entered");
15         }
16     }
17 }
```

## Output

```
PS E:\codes\java\lab 3> cd "e:\codes\java\lab 3\" ; if ($?) { javac lab3_
Enter the number of the month:
7
Respective month is : JULY
PS E:\codes\java\lab 3>
```

- 4) Write a program to find the sum of all integers greater than 40 and less than 250 that are divisible by 5.

### Code

```
lab3_ques4.java > lab3_ques4 > main(String[])
1  public class lab3_ques4{
    Run | Debug
2      public static void main(String[] args) {
3          int sum = 0;
4          int i;
5          for(i = 41; i<250; i++) {
6              if(i%5==0) {
7                  sum = sum + i;
8              }
9          }
10         System.out.println("sum is = "+ sum);
11     }
12 }
```

### Output

```
PS E:\codes\java\lab 3> e.; cd 'e:\codes\java\lab 3'; &
t' 'C:\Program Files\Eclipse Foundation\jdk-11.0.12.7-hotspot\bin\java.exe' -Xmx1024m -Xms128m -Djava.awt.headless=true -Djava.io.tmpdir=C:\Users\user\workspaceStorage\3fbf40a9e7e1d5f010fd487cbc73b17a\workspace\lab 3\temp -Djre.home=C:\Program Files\Eclipse Foundation\jdk-11.0.12.7-hotspot\bin\java.exe -cp .\bin\java.exe .\bin\java.exe
sum is = 5945
PS E:\codes\java\lab 3>
```

## Git Commands

1. **git init** - It is used to create a new Git repository. It can be used to convert an existing, unversioned project to a Git repository or initialize a new, empty repository.

```
C:\Users\Sarvagya Gupta>cd E:\  
  
C:\Users\Sarvagya Gupta>cd ..  
  
C:\Users>E:\  
'E:\' is not recognized as an internal or external command,  
operable program or batch file.  
  
C:\Users>E:  
  
E:\>cd E:\codes\java\lab 3  
  
E:\codes\java\lab 3>git init  
Initialized empty Git repository in E:/codes/java/lab 3/.git/
```

2. **git status** - The git status command displays the state of the working directory and the staging area. It tells us about the changes that have been staged, which haven't, and which files aren't being tracked by Git.

```
E:\codes\java\lab 3>git status  
On branch master  
  
No commits yet  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    lab3_ques1.class  
    lab3_ques1.java  
    lab3_ques2.class  
    lab3_ques2.java  
    lab3_ques3.class  
    lab3_ques3.java  
    lab3_ques4.class  
    lab3_ques4.java  
  
nothing added to commit but untracked files present (use "git add" to track)
```

3. `git add .` OR `<FILENAME>` - The `git add` command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit.

```
E:\codes\java\lab 3>git add .

E:\codes\java\lab 3>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   lab3_ques1.class
    new file:   lab3_ques1.java
    new file:   lab3_ques2.class
    new file:   lab3_ques2.java
    new file:   lab3_ques3.class
    new file:   lab3_ques3.java
    new file:   lab3_ques4.class
    new file:   lab3_ques4.java
```

4. `git commit -m "Some Comment"` - It records changes to the repository. Users can also add comments while saving their files.

```
E:\codes\java\lab 3>git commit -m "oops lab3 activity "
[master (root-commit) 1ff2da9] oops lab3 activity
8 files changed, 77 insertions(+)
create mode 100644 lab3_ques1.class
create mode 100644 lab3_ques1.java
create mode 100644 lab3_ques2.class
create mode 100644 lab3_ques2.java
create mode 100644 lab3_ques3.class
create mode 100644 lab3_ques3.java
create mode 100644 lab3_ques4.class
create mode 100644 lab3_ques4.java
```

5. git log - The git log command shows a list of all the commits made to a repository.

```
E:\codes\java\lab 3>git log
commit 1ff2da936faf9c28eb01c626f377c6277eefefe6 (HEAD -> master, origin/master)
Author: SarvagyaGupta-06 <sarvagya2@gmail.com>
Date: Mon Aug 30 20:39:24 2021 +0530

oops lab3 activity
```


6. git diff - This command shows the changes between commits, working tress, etc.
7. git reflog - Reflogs track when Git refs were updated in the local repository.
8. git branch - This command allows you to create, list rename, and delete branches.
9. git checkout -b //CREATE A NEW BRANCH - The git checkout command lets you navigate between the branches created by the git branch.
- 10.git remote add origin [https://github.com/SarvagyaGupta-06/oops\\_lab3.git](https://github.com/SarvagyaGupta-06/oops_lab3.git) - When you clone a repository with git clone, it automatically creates a remote connection called origin pointing back to the cloned repository.

```
E:\codes\java\lab 3>git remote add origin https://github.com/SarvagyaGupta-06/oops_lab3.git
error: remote origin already exists.
```


- 11.git clone [https://github.com/SarvagyaGupta-06/oops\\_lab3.git](https://github.com/SarvagyaGupta-06/oops_lab3.git) - git clone is a Git command-line utility that is used to target an existing repository and create a clone, or copy of the target repository.
- 12.git push origin master - git push -u origin master is used for pushing local content to GitHub.


```
E:\codes\java\lab 3>git push -u origin master
Everything up-to-date
Branch 'master' set up to track remote branch 'master' from 'origin'.
```


## Github Programmes

 SarvagyaGupta-06 / **oops\_lab3**

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

 master ▾



 1 branch









 0 tags

[Go to file](#)

[Add file ▾](#)

[Code ▾](#)

 **SarvagyaGupta-06** oops lab3 activity 1ff2da9 14 hours ago  1 commit

 lab3_ques1.class	oops lab3 activity	14 hours ago
 lab3_ques1.java	oops lab3 activity	14 hours ago
 lab3_ques2.class	oops lab3 activity	14 hours ago
 lab3_ques2.java	oops lab3 activity	14 hours ago
 lab3_ques3.class	oops lab3 activity	14 hours ago
 lab3_ques3.java	oops lab3 activity	14 hours ago
 lab3_ques4.class	oops lab3 activity	14 hours ago
 lab3_ques4.java	oops lab3 activity	14 hours ago

Help people interested in this repository understand your project by adding a README.

[Add a README](#)