Program 1

```
sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~
$ mkdir prg1

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~
$ cd prg1

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1
$ git init
Initialized empty Git repository in C:/Users/sundarapandiyan/prg1/.git/

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (master)
$ touch my_file.txt

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (master)
$ git add .

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (master)
$ git commit -m "your commit message here"
[master (root-commit) db9d5c0] your commit message here
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 my_file.txt
```

Program 2

```
sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (master)
$ git checkout master
Already on 'master'

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (master)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (feature-branch)
$ git add .

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (feature-branch)
$ git commit -m "your commit message for feature-branch"
On branch feature-branch
nothing to commit, working tree clean

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (feature-branch)
$ git checkout master
Switched to branch 'master'

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (master)
$ git merge feature-branch
Already up to date.
```

Program 3

```
sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (master)
$ mkdir stash_demo

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1 (master)
$ cd stash_demo

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ git init
Initialized empty Git repository in C:/Users/sundarapandiyan/prg1/stash_demo/.git/

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ touch mystash.txt

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ echo "print (' hello world') " >mystash.txt

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ git add mystash.txt
warning: LF will be replaced by CRLF in mystash.txt.
The file will have its original line endings in your working directory

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ git add . mystash.txt

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ git commit -m " Initial commit : add main .txt"
[master (root-commit) d98e38a]  Initial commit : add main .txt
 1 file changed, 1 insertion(+)
 create mode 100644 mystash.txt

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ echo "print ('this is a stash demo ')"> mystash.txt

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   mystash.txt

no changes added to commit (use "git add" and/or "git commit -a")

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
```

```
sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ git stash save "your stash message"
warning: LF will be replaced by CRLF in mystash.txt.
The file will have its original line endings in your working directory
Saved working directory and index state On master: your stash message

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ git stash list
stash@{0}: On master: your stash message

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ git checkout -b target branch
fatal: 'branch' is not a commit and a branch 'target' cannot be created from it

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (master)
$ git checkout -b target-branch
Switched to a new branch 'target-branch'

sundarapandiyan@DESKTOP-KV3ENAQ MINGW64 ~/prg1/stash_demo (target-branch)
$ git stash apply
On branch target-branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   mystash.txt

no changes added to commit (use "git add" and/or "git commit -a")
```