# <u>How to fetch file through GITHUB in yocto -project.</u>

First write the recipe file(.bb)  to configure the file through git hub.
Cretae a new directory in your creating layer (my-layer).
Create a directory in this layer.
This is the path where is create your package recipes.

```
/home/arjun/YOCTO_PROJECT/poky/meta-txt/recipes-example/gittest/gittest.bb
$ mkdir gittest
```

Enter this directory and create one recipe or .bb file to fetch the code through github.

```
$ cd gittest
$ vi gittest.bb
```

Enter this data in this file(**gittest.bb**)

```
DESCRIPTION = "A simple Hello World program which is fetch through git"

LICENSE = "MIT"

LIC_FILES_CHKSUM =
"file://${COREBASE}/meta/COPYING.MIT;md5=3da9cfbcb788c80a0384361b4de20420"

SRCREV = "22153a68a563d7fb0427c662126fdf838fbd13af"

SRC_URI[sha256sum] =
"e42ca16f10bcb4f303adfc550c08718e287d4b5b2e3eff1ee6847f57a49942b8"

SRC_URI = "https://raw.githubusercontent.com/akshayakshay30/Hello-world.c/master/
factorial.c;branch=master"

S = "${WORKDIR}"

RDEPENDS:${PN} = ""

do_compile() {
   ${CC} ${CFLAGS} ${LDFLAGS} factorial.c -o factorial
}

do_install() {
   install -d ${D}${bindir}
   install -m 0755 factorial ${D}${bindir}/factorial
}

FILES_${PN} = "{bindir}/factorial"
```

save and exit. The file

basic terms understand for this file.

[**https://raw.githubusercontent.com/akshayakshay30/Hello-world.c/master/factorial.c;branch=master**](https://raw.githubusercontent.com/akshayakshay30/Hello-world.c/master/factorial.c;branch=master)

Note :: this link is not the acutal link for factorial.c file. So this is raw file link , because all the have been uploaded in the form of raw in the git. So it requirred raw type link to directly clone this original file. Otherwise you are add normal file link then is show ERROR  to compile this file. It creates the file and not original c file it save c file in raw type object, measn it include many object file in this c file. This file is not compile because this is not a c file. It is requirrred to use only raw type file.

**SRCREV = "22153a68a563d7fb0427c662126fdf838fbd13af"**
This is commit id. Every program have uniquie commit id. It is basically used only for actual program id or diretly fetch this file.

All terms and variables are common.

Go to the Conf folder of your layer and open layer.conf file and add path of your layer to configure directly.
**$ vi layer.conf**
**BBFILES += "${LAYERDIR}/recipes-example/githello/gittest.bb"**

Now set the build environment.

**$ source oe-init-buil-env ../build**

this command helps to directly go to the build folder.
In build folder enter conf folder and open local.conf file.

**$ cd / build / conf**
**$ vi local.conf**

and add this line at last of the this file.

**IMAGE_INSTALL:append = " gittest"**
save and exit the file.

And compile first pacakge using this command.
**$ bitbake gittest**

And compile final build image.
**$ bitbake core-image-minimal**

Now check this image output through qemu beacause iam using qemux86-64 machine.
**$ runqemu qemux86-64  nographic**

Your output will be locate in the **/ usr /bin / factorial** >>> factorial this is binary format.