1. **Refer the code snippet below and answer the queries.**

int g\_value =10; /\* global variable \*/

int main()

{

int pid;

int l\_value =5;

printf(“Writing a sample code\n”);

pid = fork(); /\* fork() returns 0 to child process and process id of child to parent process\*/

if(0 == pid)

{

printf(“For child Local variable value=%d\n and global variable value=%d\n”,l\_value,g\_value);

exit(0);

}

else

{

printf(“For parent Local variable value=%d\n and global variable value=%d\n”,l\_value,g\_value);

}

printf(“Code common for both parent process and child process\n);

return 0;

}

1. **What will be the output of parent process and child processes?**

Output: Writing a sample code For parent Local variable value= 5 and global variable value= 10 Code common for both parent process and child process For Child Local variable value= 5 and global variable value= 10

1. **Find out whether the value of local variable and global variable value will be same for both parent process and child process**

Yes, the value of local variable and global variable value will be same for both parent process and child process.

1. **Will the order of execution be same always or could be different? Will it impact the output?**

No, the order of execution won’t be same always, it could be different but the values of variable will always be same**.**

1. **Why the first printf() statement will be executed only by parent process and not by child process?**

The first printf() statement will be executed only by parent process and not by child process because fork() was called after it, so the child will only copy the statements after the fork() called.