Flipped Learning Pre-class 점검 문제		주차	10
교과목명	오픈소스소프트웨어	교수	강환수
주제	병합과 충돌 해결		

## □ 객관식

- 1. 브랜치 main에서 브랜치 feature를 병합하는 명령어는? (4)
- ① \$ git merge main

2 \$ git main merge feature

3 \$ git merge main feature

4 \$ git merge feature

2. 병합할 브랜치의 조상(ancester) 중에 위치한 브랜치에서 기본으로 병합하는 방식은?

(4)

① 3-way

② squash

③ rebase

4 fast-forward

3. 다음 중 병합할 브랜치의 조상(ancester) 중에 위치한 브랜치에서 3-way 병합을 수행하는 명령의 옵션은? (3)

① --non-ff

② --ff

③ --no-ff

4 -- squash

- 4. 다음 병합에 대한 설명으로 바르지 않은 것은? (2)
- ① fast-forward, 3-way 등의 병합 방식이 있다.
- ② 병합은 항상 성공한다.
- ③ 3-way 병합에서는 새로운 커밋이 생성된다.
- ④ fast-forward 병합에서는 새로운 커밋이 생성되지 않는다.
- 5. 다음 중 병합 충돌에서 병합을 취소하는 명령어는? (1)
- ① \$ git merge --abort

② \$ git merge abort

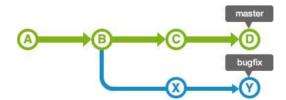
③ \$ git abort merge

4 \$ git merge --cancel

#### □ 주관식

다음 master에서 bugfix를 3-way 병합하는 명령어는?

\$ git merge bugfix



e-learning	콘텐츠 자기주도학습 점검 문제	주차	11
교과목명	오픈소스소프트웨어	교수	강환수
주제	rebase와 충돌 해결, 커밋 수정		

# □ 객관식

- 1. 베이스를 바꾸는 재배치 명령어는? (2)
- ① \$ git merge feature
- ② \$ git rebase main
- ③ \$ git 3-way-merge main
- 4 \$ git base main
- 2. 명령 rebase에서 --interactive에서 사용하는 명령이 아닌 것은? (3)
- ① squash

2 pick

3 cancel

- 4 reword
- 3. 명령 rebase에서 이미 수행한 커밋을 수정하는 옵션은? (1)
- ① --interactive

② --amend

③ --no-edit

- ④ --i
- 4. 다음 병합에 대한 설명으로 바르지 않은 것은? (1)
- ① rebase에서 fast-forward는 자동으로 수행된다.
- ② 명령 rebase에서 브랜치 2개가 지정될 수 있다.
- ③ rebase에서 충돌이 나면 옵션 --abort로 병합을 취소할 수 있다.
- ④ rebase에서 충돌이 나면 해결한 후 옵션 --continue로 병합할 수 있다.
- 5. 이미 수행한 커밋에서 마지막 커밋 메시지를 바로 'last'로 수정하는 명령은? (4)
- ① \$ git commit 'last'

② \$ git commit -amend 'last'

③ \$ git commit --last 'last'

④ \$ git commit --amend -m 'last'

## □ 주관식

다음 결과가 나오도록 하는 rebase 명령어는?

\$ git rebase master bugfix





e-learning	콘텐츠 자기주도학습 점검 문제	주차	13
교과목명	오픈소스소프트웨어	교수	강환수
주제	되돌리기 reset과 revert		

# □ 객관식

- 1. 다음 중 reset의 주요 옵션이 아닌 것은? (4)
- ① --soft

② --mixed

③ --hard

- 4 --head
- 2. 명령 'git reset HEAD~'와 같은 명령은? (1)
- ① \$ git reset --mixed HEAD~
- ② \$ git reset --soft HEAD~
- 3 \$ git reset --hard HEAD~
- 4 \$ git revert HEAD~
- 3. 명령 reset 전의 커밋을 참조하는 이름은? (4)
- ① STASH@{1}

② HEAD

③ HEAD~

- 4 ORIG\_HEAD
- 4. 명령 git revert HEAD --no-edit 이후 작업 디렉토리와 스테이지 영역, 커밋 내용이 동 일한 명령은? (2)
- ① \$ git reset --hard HEAD
- 2 \$ git reset --hard HEAD~
- ③ \$ git reset --mixed HEAD~ ④ \$ git reset --soft HEAD~
- 5. 특정 커밋 이력을 취소하고 바로 이전 상태로 돌아가는 명령은? (3)
- 1) rebase

2 reset

3 revert

4 checkout

### □ 주관식

다음 reset 명령의 옵션을 기술하시오.

