Drasil Future Contributors' Test
McMaster University
Faculty of Engineering, Department of Computing and Software

Answer Key: Large arrow ($\Longleftarrow$) for correct

Future Drasil Contributors, **Version 1** Dr. S. Smith
DURATION OF TEST: 1.5 hours - 2.0 hours (recommended)
MCMASTER UNIVERSITY TEST

Please CLEARLY print:

NAME:

This test paper includes **??** pages and 22 questions. You are responsible for ensuring that your copy of the test paper is complete. Bring any discrepancy to the attention of the person administering this test.

1–17		17
18–22		10
Total		27

Question 1 [**1 mark**] What command is used to create a local version of a remote repository?

A. `git copy <link to repo>`

B. `git clone <link to repo>` ⟸

C. `clone <link to repo>`

D. `copy <link to repo>`

Explain your answer here.

ANSWER:
Sample Answer: `git clone <link to repo>` clones the remote repo and stores it locally

Question 2 [**1 mark**] The `git status` command displays:

A. verified status of all repo files

B. list of last 5 commits to the repo

C. paths with differences between current state of repo and last commit ⟸

D. changes made to files after last commit

Explain your answer here.

ANSWER:
Sample Answer: `git status` shows all files with changes made after previous commit/push

Question 3 [**1 mark**] If you are a Windows OS user working with Git Bash App, always run the command each time you open an instance of git bash:

A. `chcp.com 65001`

B. `set encoding=utf-8`

C. `chcp 65001`

D. `A. or C.` ⟸

Explain your answer here.

ANSWER:
Sample Answer: `chcp.com 65001` or `chcp 65001` set terminal encoding to UTF-8 (this allows Drasil to handle Greek letters/unicode symbols)

Question 4 [1 mark] What is the correct order of the following `git` commands?

A. `git push, git add, git clone, git commit`

B. `git add, git clone, git push, git commit`

C. `git clone, git add, git push, git commit`

D. `git clone, git add, git commit, git push` ⟸

Explain your answer here.

ANSWER:
Sample Answer: `git clone` creates local copy of remote repo, `git add` stages changed files, `git commit` commits staged files, and `git push` pushes committed files (changes) to remote repo

Question 5 [1 mark] Which of the below phrases can you use to link a relevant issue to a pull request on GitHub, without closing the issue once the PR is merged? (#HASH is the issue hash #)

A. closes #HASH

B. contributes to #HASH ⟸

Explain your answer here.

ANSWER:
Sample Answer: using 'contributes to #HASH' in the PR comment links the relevant issue by mentioning it; 'closes' is a keyword that tells GitHub to close this issue once this PR is merged

Question 6 [1 mark] The preferred coding style describes how lines should not be more than ___ characters wide.

A. 60

B. 90

C. 80 ⟸

D. 70

Explain your answer here.

ANSWER:
Sample Answer: the preferred coding style (refer to Contributor's Guide) is to avoid making 'long lines' more than 80 characters wide

Question 7 [**1 mark**] When making pull requests involving changes to multiple files (e.g. Haskell scripts and stable folder files), remember to:

A. use multiple `git add` to stage all changed files before doing a single commit ⟸

B. update the stable files first and push them, then repeat with the scripts using `git add`

C. only update all scripts in one commit using multiple `git add`

D. only update all changed 'stable' folder files in one commit using multiple `git add`

Explain your answer here.

ANSWER:
Sample Answer: it is important to stage **all** changed files (stable and code) using multiple `git add` before doing a single commit to avoid Continuous Integration build errors

Question 8 [**1 mark**] The `git branch` command:

A. shows the items under your current branch on your local repo

B. shows a list of all your current branches on your local repo ⟸

C. shows branches dependent on your current branch on your local repo

D. shows a list of all current branches on the remote repo

Explain your answer here.

ANSWER:
Sample Answer: `git branch` is used to display all current branches on your local repo (as a list)

Question 9 [**1 mark**] When closing an issue, please provide:

A. Rationale

B. Relevant Links to other related issues

C. Linked Pull Requests

D. Any or all of the above ⟸

Explain your answer here.

ANSWER:
Sample Answer: any or all of the above help other contributors understand why the issue is being closed/direct them to any changes made to address those issues (i.e. give further clarification)

Question 10 [1 mark] To **only** build the 2D Rigid Body Physics Library example (gamephysics_diff) using the Drasil framework, run the command:

A. `setup gamephysics_diff`

B. `make gamephysics_diff` ⟸

C. `stack exec gamephysics_diff`

D. `make`

Explain your answer here.

ANSWER:
Sample Answer: `make gamephysics_diff` directs Drasil to **only** make the 2D Rigid Body Physics Library example; in contrast, `make` directs Drasil to make **all** examples

Question 11 [1 mark] The `git pull` command is used to:

A. sync your local version with remote version of the repo ⟸

B. displays changes made to the remote version of the repo

C. sync remote version with your local version of the repo

D. updates the remote repo with other people's changes

Explain your answer here.

ANSWER:
Sample Answer: `git pull` retrieves all changes made to the remote repo, and updates the local repo

Question 12 [1 mark] To run the Glass-BR example (glassbr) using the Drasil framework (assume that the example has already been built), run the command:

A. `make glassbr`

B. `exec glassbr`

C. `stack glassbr`

D. `stack exec glassbr` ⟸

Explain your answer here.

ANSWER:
Sample Answer: `stack exec` is used to run a specific example that has already been built, using stack

Question 13 [1 mark] What is the difference between a remote branch and a local branch?

A. a local branch is stored on a server; a remote branch is stored on your computer

B. a remote branch is stored on a server; a local branch is stored on your computer ⟸

C. both remote and local branches are stored on a server

D. both remote and local branches are stored on your computer

Explain your answer here.

ANSWER:
Sample Answer: A remote branch is stored on the (GitHub) server. A local branch is a copy of the remote branch stored on your computer. The local branch is where all editing takes place, and those changes are pushed to the remote branch.

Question 14 [1 mark] What is `origin`?

A. the latest branch created

B. a basic template branch

C. the master branch ⟸

D. none of the above

Explain your answer here.

ANSWER:
Sample Answer: `origin` is the original or `master` branch into which all changes made on all other branches are made into. It is the master copy of a remote repo.

Question 15 [1 mark] What is `HEAD`?

A. the latest commit in your current branch

B. the base/master branch of your remote repo

C. the active item referenced by your current repo

D. A. and C. ⟸

Explain your answer here.

ANSWER:
Sample Answer: `HEAD` is the active commit that your repo is currently on (usually your latest commit).

Question 16 [**1 mark**] Which branch are you pulling from when you execute the following commands:

```
git branch
git switch master
git switch sample
git pull
```

A. master

B. sample ⟸

C. both master and sample

D. neither master nor sample

Explain your answer here.

ANSWER:
Sample Answer: `git branch` displays a list of all existing branches on the current repo.
`git switch master` and `git switch sample` switch the active branch to master, and then to sample.
`git pull` pulls from and updates the active branch, which would be sample.

Question 17 [**1 mark**] Suppose that your have created a new branch on your local repo **only**. Which of the following commands would you use to push your new branch onto the remote repo?

A. `git push --set-upstream <new branch name>`

B. `git push`

C. `git push <new branch name>`

D. `git push --set-upstream origin <new branch name>` ⟸

E. none of the above

Explain your answer here.

ANSWER:
Sample Answer: the command `git push --set-upstream origin <new branch name>` is used to push your local branch if a corresponding branch **does not** already exist on the upstream repo. If a corresponding branch **already** exists on the remote repo, then use the command `git push`.

Question 18 [2 marks] Pretend that you have made a commit on your local repo that you would like to undo. Describe the process that you would use to undo the commit if (choose one to answer):

A. your commit is only on your local repo

B. your commit has also been pushed to the remote repo

Be sure to include any commands that you use to accomplish this task.

Answer here (please indicate which option you chose).

ANSWER:
Sample Answer: (option A)

- use `git log <branchName> --oneline` to view a simplified log of all commits made and their accompanying commit hashes (prefixed alphanumeric string)
- use `git checkout <#commitHash>` to view the specific commit in more detail (this command matches your working directory to the exact state of the #commitHash commit)
- use `git checkout <branchName>` to get back to the branch (to undo the commit #commitHash)
- use `git reset --hard <#commitHash>` to reset the commit history, removing the commit #commitHash from the `git log` output completely

Sample Answer: (option B)

- repeat the same first three steps described in the sample answer for option A
- use `git revert <#commitHash>` to revert the specified commit on the repo, by creating a new (inverse) commit that undoes what commit #commitHash did

Question 19 [2 marks] When creating a new issue on GitHub, describe two tips to follow that help to ensure that the new issue includes enough information (context).

Answer here.

ANSWER:
Sample Answers (any two of the following work):

- include (annotated) excerpts of PDF/HTML documentation when referring to output (desired or generated); highlighting specific portions of such relevant screenshots helps important info stand out to the reader
- linking related issues, pull requests, comments and commit hashes; provides easy navigation through related content and discussion significant to the issue
- creating permalinks (permanent links) to code sections/snippets relevant to the issue
- directly inserting code into Issue comments (using markdown format); useful for recommending/proposing solutions of multi-line code (mentioning the language allows for syntax highlighting, which is helpful)

Question 20 [2 marks] Explain the concept of cherry-picking in GitHub. Be sure to include what command(s) and information you would use to accomplish it.

Answer here.

ANSWER:
Sample Answer:

- cherry-picking in GitHub is the concept/process of moving a specific commit from one branch onto another (new branch)
- there are two commands you would likely need to use:
  - `git checkout` to ensure that you are on the branch you want to commit to
  - `git cherry-pick <commit-hash>` to move a specific commit (referenced by commit-hash) to the new branch
- the information you would need to accomplish cherry-picking is the `commit-hash` of the specific commit you would like to move to the new branch

Question 21 [2 marks] Describe Drasil (what is it, what does it do), and discuss its main goals.

Answer here.

ANSWER:
Sample Answer:

- Drasil is a framework for generating **all** software artifacts (e.g. code, documentation) from a stable knowledge base
- the main goals of Drasil are to:
  - reduce knowledge duplication; concepts, ideas and algorithms should only have to be defined once (and be reused easily)
  - improve traceability; more easily display how various components of the program are linked/dependent upon each other
  - improve maintainability, verifiability, traceability and other software qualities (more as a side-effect of methods used rather than as a main goal)

Question 22 [2 marks] Imagine that you have made changes to some files in the Drasil code on your local repository. Describe the process that you would take to update the remote repository for Drasil with your changes. List any commands that you would use here as well.

Answer here.

ANSWER:
Sample Answer:

- `git add <relative path to file>`: stage a file to a commit; be sure to repeat this multiple times for multiple changed files, or use `git add *` cautiously (command stages all changed files)
- `git commit -m "commit message"`: commit your staged changes to your local repository; be sure to include a descriptive commit message
- `git push`: push changes from your local repository to the remote repository (thus updating it)