The TRAC program

• Fermilab’s Teacher Research Associates (TRAC) program provides outstanding science, mathematics, computer science (CS) and technology teachers with professional summer research experiences in science, engineering or technology.

• Increased teacher awareness and understanding of cutting-edge science and technology is then transferred to the classroom. Teachers are assigned Fermilab mentors and jobs that best match their skills and interests.

• The TRAC program is funded by a Department of Energy grant.
Agenda

• Introduction
• TARGET program
• GlideinWMS Project
Presenter – A. Debarlaben

- BSEE from Ohio State University
- Masters in CS from IIT
- Worked for over 18 years as a Software Engineer (GE, Northrop Grumman, Lucent Technologies)
- Certified to teach Math and CS

- VP of Chicago Suburban Computer Science Teacher Association (CSTA)
- State of Illinois Finalist for the Presidential Awards for Excellence in Math and Science Teaching (PAEMST)
Marco works here at Fermilab in the Scientific Computing Division. He is the project and technical lead for GlideinWMS, a tool used to provision the computing resources for CMS and most Fermilab experiments. Before that, he worked on the design and development of other distributed software projects at Fermilab and the University of Chicago.

Marco is a TRAC Mentor and the Python Instructor for the TARGET program. In addition to working on a project, I supported the TARGET program.
The TARGET program

• The program is a highly competitive six-week summer internship opportunity for Illinois high school sophomores and juniors who have strong interest and demonstrated aptitude for physics, mathematics, computer science and engineering.

• The program encourages and aims to increase the representation of underrepresented minorities (Black, Hispanic/Latino, Hawaiian/Pacific Islander, Alaska Native/American Indian) and women in the sciences and engineering at the college level and consequently the workforce.
TARGET Program
Python Class

• Discussed lessons with instructor

• Answered Python questions during class

• Created Kahoot games for review and to engage students
A Python example using the CoLaboratory IDE

6. Flow Control

6.1 If Statement

The if statement is very useful for running tests and doing actions based on those tests results. Here is an example of this in practice:

```python
my_age = input("What is your age?") # input is a statement that displays text & takes an input
if int(my_age) >= 16:
    print("You can drive with a license")
elif int(my_age) == 15:
    print("You can drive with a parent")
```

What is your age? 21
You can drive with a license
Learning Git and GitHub to help manage the development environment
Goal of the Project

- To implement structured logging in GlideinWMS and analyze the data produced to gain more insights on the system.

- To learn about distributed computing systems, deploy services using virtual machines (OpenStack) and containers (Docker/Podman), implement logging solutions and analyze aggregated data to optimize security and performance.
Purpose of the project

• To move to a structured format of logging to enable better identification of the logged information, easy understanding of system specifics and problems when encountered while improving security and easing aggregation and data analyses.

• Additionally, having this new, structured logging format would also ease the integration with HEPCloud as it already uses a structured format for logging.
Logging Code

**Standard Python Logging**
Present logic in log modules

```python
import logging

logSupport.log = logging.getLogger("frontend")
```

**Structured Python Logging**
Create a function to add logging transparency

```python
import structlog

def getLogger(name):
    return structlog.getLogger(name)

Modify logic in log modules

```python
logSupport.log = logSupport.getLogger("frontend")
```
Comparison of Structure vs Standard Log Output

```python
import structlog
import logging

logger_structlog = structlog.get_logger()
logger_standard = logging.getLogger(__name__)
logging.basicConfig(level=logging.INFO)

logger_structlog.info(
    "Hello Pythonista!",
    key_id="1234",
    conference_name="EuroPython",
    talk_name="Can we deploy yet?",
)

logger_standard.info(
    "Hello Pythonista! Conference name %s, talk name %s, key_id = %s"
    % ("EuroPython", "Can we deploy yet?", "1234")
)

As a result, we will get:

2028-07-19 22:07:23 Hello Pythonista! conference_name=EuroPython key_id=1234 talk_name=Can we deploy yet? INFO:_main__:Hello Pythonista! Conference name EuroPython, talk name Can we deploy yet?, key_id = 1234
```
What is GlideinWMS?

- GlideinWMS is a workload management system for experiments including Compact Muon Solenoid (CMS) and Deep Underground Neutrino Experiment (DUNE).

- The GlideinWMS framework produces a large amount of logging information, which is useful for individual service operators.
Summary

Work with GlideinWMS and HEPCloud teams and domain experts, in a collaborative environment, with state-of-the-art scientific distributed computing resources, to solve problems and implement solutions, with a concrete impact on the scientific community.
Acknowledgements and References

• This work was done under the Teacher Research Associates (TRAC) Program at Fermilab.

• This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

• GlideinWMS slides are based in part on previous presentations by the GlideinWMS project developers

• Git and GitHub management photo taken from blog written and created by Molly Nemerever

• Python and CoLaboratory image taken from Marco Mambelli | GlideinWMS introduction presentation
Take Away as a Teacher

I gained experience with:

• CoLaboratory
• Git and GitHub
• Python
• Logging as a possible debugging tool

Thank You

Anita F. Debarlaben
afdebarlaben@gmail.com
QUESTIONS???