Class Activity 1.2: Variables and Arithmetic Operators

In this activity, we will review variables and arithmetic operators in C++.

Please fill in the roles for each member of your team. Take a look at the description of each role to see its responsibilities. In case there are only three people in the group, please assign the same person to the **Presenter** and **Reflector** role. It is a good idea to select roles that you have not recently taken.

Team name: _____ Date: _____

Role	Team Member Name
Manager. Makes sure team starts quickly and remain focused during the activity; takes care of time management; makes sure all voices in the team are heard	
Presenter. The only person in the team assigned to communicate questions and clarifications with the instructor or other teams; ensures all team members have had a chance to respond before asking outside sources; ensures that everyone in the team agrees on what to ask if an outside source is needed; presents conclusions of the team to the class, as requested.	
Reflector. Guides consensus-building process so that the team agrees on responses to questions; observes team dynamics and behavior with respect to the learning process; reports to the team periodically during the activity on how the team performs; possibly report to the entire class about how well the team is operating.	
Recorder. Records the names and roles of the group members at the beginning of each activity; records the important aspects of group discussions, observations, insights, etc.; the recorder's report is a log of the important concepts that the group has learned.	

1. What do you think will be the output of the following code?

Code:
```
#include <iostream>
int main() {
  int val;
  std::cout << val;
  return 0;
}
```

Output: Random integer value

2. Why do you think the code will output what you answered in #1?

An uninitialized integer value will get a random value.

3. What data type do you think is most appropriate for representing the following values? Choose from *double*, *unsigned int*, *int*, *bool*, *char*, and *std::string*.

Literal	**Data type**
take note whether a message has already been sent or not	bool
a letter grade	char
person's age	unsigned int
an email address	std::string
the price of a grocery item	double
ID ranging from 0 to 5000	unsigned int

4. Match the following literals with their corresponding data type.

Identifier
`float`
`long`
`char`
`std::string`

Data type
`245L (long)`
`'A' (char)`
`"A" (std::string)`
`18.37F (float)`

5. Complete the code by providing a single statement that will display the name and age of a person given the variables used. Use the output to guide your code.

Output:

```
Name: Tuffy
Age: 22
```

Code:

```
#include <iostream>
int main() {
  std::string name = "Tuffy";
  unsigned int age = 22;
  // Provide code here
  std::cout << "Name: " << name << "\n";
  std::cout << "Age: " << age << "\n";


  return 0;
}
```

6. What is wrong with the code below?

Code:

```
int main() {
  int num1 = 5;
  std::cout << num1;
  std::cout << num2;
  int num2 = 12;
  return 0;
}
```

Answer: You can't use a variable before it is declared.

7. What would be the result of the following operation: 15/2 ?

Answer:

7

8. What would be the result of the following operation: 15.0/2 ?

Answer:

7.5

9. Write a complete C++ program that uses variables to store the name of the user, the user's ID, and the user's restaurant bill. Initialize the name of the user to a group member's name, their ID to any positive 5 digit number, and the restaurant bill to 30 (no need to get user input). Use *std::cout* to display the values of all variables on the screen. You can freely design your screen output.

Possible output:

```
Name: Tuffy Titan
ID: 12345
Bill: $30.00
```

Code:

```
#include <iostream>
#include <iomanip>
int main() {
  unsigned int id = 12345;
  std::string name = "Tuffy Titan";
  double bill = 30.0;
  std::cout << "Name: " << name << "\n";
  std::cout << "ID: " << id << "\n";
  std::cout << std::fixed
            << std::setprecision(2)
            << "Bill: " << bill << "\n";
  return 0;
}
```