NAME: TANKISO MASOEBE

PROGRAMMING WITH C++ WEKK 3

1.SCENARIO ANALYSIS:

In a budget tracker program, primitive data types like int and double play an important role in managing memory and calculation accuracy. An int is used for whole numbers such as the number of transactions or the month index. It occupies less memory and processes faster, making it efficient for values that do not require decimals. A double is used for monetary values like income, expenses, and remaining balance because it supports decimal precision, which is essential for accurate financial calculations. However, double uses more memory than int, so it should only be used where precision is necessary. The const keyword is applied to fixed values such as tax rates or monthly budget limits to prevent accidental modification. Using const improves program safety, readability, and reliability. Together, these choices balance memory efficiency, precision, and data protection in the budget tracker.

2.CONCEPT RESEARCH:

Signed and unsigned data types differ in how they handle numeric values and arithmetic operations. Signed types, such as int, can store both positive and negative numbers, making them suitable for calculations like profit and loss. Unsigned types, such as unsigned int, store only non-negative numbers, allowing a larger maximum value using the same memory size. However, unsigned arithmetic can cause unexpected results. For example, if unsigned int x = 0; then x - 1 results in a very large number due to underflow. Signed types are safer for general arithmetic, while unsigned types are best for counts and sizes.

3.TOOL PRACTICE :

Valgrind is a useful debugging tool for detecting memory-related errors such as uninitialized variables. When running Valgrind on sample code containing uninitialized variables, it reports warnings like "use of uninitialized value." These issues can lead to unpredictable program behavior, incorrect calculations, or crashes because the variable may contain random memory data. Valgrind helps developers identify exactly where the problem occurs, making it easier to fix by initializing variables before use. Detecting these issues early improves program reliability, accuracy, and security. Overall,

Valgrind reduces debugging time and ensures better memory management, especially in larger or more complex C or C++ programs.

4.APLICATION PRACTICE

In the fictional game design, a constant palette defines fixed values such as player movement speed, jump height, and attack cooldown. These constants ensure consistent gameplay and prevent accidental changes during execution. Variable stats include health, stamina, experience points, and score, which change based on player actions and game events. In Draw.io or on paper, constants are labeled separately from variables to show their roles clearly. During integration, constants are used by the game engine to control core mechanics, while variables update in real time as the player progresses. This separation improves balance, readability, and easier maintenance of the game code.