**A close up of a sign

Description automatically generated**

**AMERICAN INTERNATIONAL**

**UNIVERSITY-BANGLADESH** 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: **Asmaul Hossain Akash**

Student ID: **20-44209-3**

Program**: BSc (CSE)**

Semester: **Fall 20-21**

Section: **B1**

Submission Date: **October 25, 2020**

Assignment Title: **Describing an Entire Code System**

Assignment No: **01**

Course Title: **Introduction to Programming**

Course Code: **00152**

Course Teacher: **Aneem Al Ahsan Rupai**

*Code:*

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  float phy, chem, bio, per, sum;  phy = 91.00;  chem = 87.45;  bio = 51.00;  sum = phy+chem+bio;  cout<<"Total Marks: "<<sum<<endl;  per = (sum\*100)/300;  cout<<"The student obtained "<<per<<"% marks";  } |

*Output:*

|  |
| --- |
| Total Marks: 229.45  The student obtained 76.4833% marks  Process returned 0 (0x0) execution time : 0.330 s  Press any key to continue. |

*Explanation:*

**Preprocessor Derivative**: Preprocessor directives are lines included in a code of programs preceded by a hash sign (#).

|  |
| --- |
| #include <iostream>  Here,  #**include** directive tells preprocessor to include the contents of the file specified in the input stream to the compiler and continue.  **<iostream>** is a header that defines the standard input/output of stream objects. |

**Namespace**: A namespace is a declarative sector that grants a range to the compiler (the names of types, functions, variables, etc.) inside it.

|  |
| --- |
| using namespace std;  means we use the **namespace** named **std**. (**Std** is the shortened form of standard.) A semicolon ( ; ) marks the end of a statement. |

**Function:** A function is a block of code that only runs when it is declared.

|  |
| --- |
| int main () '**int main ()**' means that our function needs to return some integer at the end of the execution. “**main**” is a special function in c++. |

**Variable:** A variable renders us with defined storage that our programs can manipulate.

|  |
| --- |
| Here **‘int,float’** are variable.  The line “**float phy, chem, bio, per, sum;**” both declares and defines the variables phy, chem , bio, per and sum; which instructs the compiler to create variables named phy, chem , bio, per and sum of type float. |

**Pointer:** A pointer is a variable that can store such an address.

|  |
| --- |
| **phy = 91.00; chem = 87.45; bio = 51.00;**  here “phy = 91.00” means initialize phy to value 91.00 and same to others. |

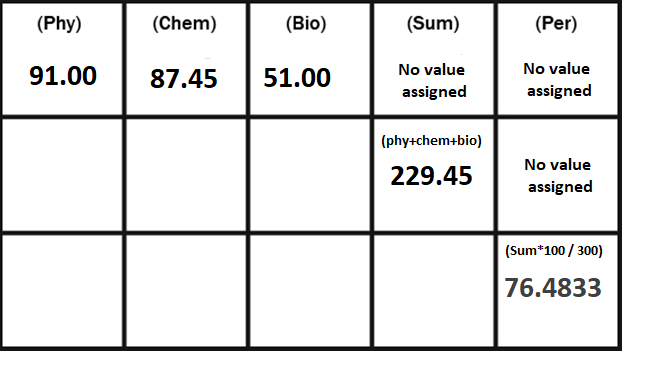
**Operators:** An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations.

|  |
| --- |
| Here, + , \* , /, << are the operators.  + means sum or addition  \* Means multiplication  / means division   << operator inserts the data that reflects it into the stream that leads it. |

**Stream:** A **stream** is a concept that renders a program on which input, and output operations are performed.

|  |
| --- |
| Here “**cout”** is a stream. Which is a standard output stream. |

**Memory Stage Diagram:**

****

**Analysis from Diagram:**

1. After getting input for phy,chem, and bio variable in float data type, These each data will be stored in a separate space in memory.

(float Phy = 91.00, chem = 87.45, bio =51.00)

1. Sum of phy, chem, and bio is a float number, and it will be stored in separate memory space.

(sum = phy + chem + bio  
 => sum = 91.00 + 87.45 + 51.00  
=> sum = 229.45)

1. Value of sum will be multiplied by 100 and then divided by 300, and it will be named as ‘per’ variable and will be stored in a separate space in memory.  
    (per = sum\*100)/300  
    => per = 229.45\*100/300  
    => per = 76.4833 )
2. After getting the printing command (cout<<"The student obtained "<<per<<"% marks";) the screen will show the result.

(The student obtained 76.4833% marks)