## Monday Group

The prices of the 5 products as well as the tax rate are constants. They can be defined in two possible ways: both are acceptable.

|  |
| --- |
| #define TAX\_RATE 0.06  #define PRICE\_PRODUCT\_1 2.98  #define PRICE\_PRODUCT\_2 4.50  .....  .....  OR  int main(void)  {  const double TAX\_RATE = 0.06  const double PRICE\_PRODUCT\_1 = 2.98  const double PRICE\_PRODUCT\_2 = 4.50  .....  .....    } |

Function 2, 3 and 4 may or may not use addresses / references. However, if the function is only returning a single value (as is the case here for function 2, 3 and 4) it is better to return the value directly rather than use an address / reference.

|  |
| --- |
| **// Example of calling functions 2, 3 and 4 where the result is returned**  **// directly. This is the preferred approach**    double calculateSubTotal(int quantity1, int quantity2, int quantity3,  int quantity4, int quantity5);  double calculateTax(double subTotal, double taxRate);  double calculateTotal(double subTotal, double tax);  ...  ...  int main(void)  {  double subTotal, tax, totalPrice;  ....  ....  subTotal = calculateSubTotal(product1\_quantity, product2\_quantity, product3\_quantity,  product4\_quantity, product5\_quantity);  tax = calculateTax(subTotal, taxRate);  totalPrice = calculateTotal(subTotal, tax);  ....  ....  }  **// Example of calling functions 2, 3 and 4 where the result is returned via a reference**    void calculateSubTotal(int quantity1, int quantity2, int quantity3,  int quantity4, int quantity5, double &subTotal);  void calculateTax(double subTotal, double taxRate, double &tax);  void calculateTotal(double subTotal, double tax, double &total);  ...  ...  int main(void)  {  double subTotal = 0.0;  double tax = 0.0;  double totalPrice = 0.0;  ....  ....  calculateSubTotal(product1\_quantity, product2\_quantity, product3\_quantity,  product4\_quantity, product5\_quantity, subTotal);  calculateTax(subTotal, taxRate, tax);  calculateTotal(subTotal, tax, totalPrice);  ....  ....  } |

The main() function can call function 2, 3 and 4 directly (easiest). Alternatively, main() can call an intermediate function A, which in turn calls function 2, 3 and 4. In this case, function A must return the results (subtotal, tax, total price) to main()

## Tuesday Group

Function 2 and 3 may or may not use addresses / references. However, if the function is only returning a single value (as is the case here for function 2 and 3) it is better to return the value directly rather than use an address / reference.

|  |
| --- |
| **// Example of calling functions 2 and 3 where the result is returned**  **// directly. This is the preferred approach**  double compute\_total\_interest(double loan, double annual\_interest, double num\_month\_payments);  double compute\_monthly\_payment(double loan, double total\_interest, int num\_month\_payments);  ....  int main(void)  {  double total\_interest, monthly\_payment;  total\_interest = compute\_total\_interest(loan, annual\_interest, num\_month\_payments);  monthly\_payment = compute\_monthly\_payment(loan, total\_interest, num\_month\_payments);  ...........  }    **// Example of calling functions 2 and 3 where the result is returned via a reference**    void compute\_total\_interest(double loan, double annual\_interest, double num\_month\_payments, double& total\_interest);  void compute\_monthly\_payment(double loan, double total\_interest, int num\_month\_payments, double& monthly\_payment);  ....  int main(void)  {  double total\_interest = 0.0;  double monthly\_payment = 0.0;  compute\_total\_interest(loan, annual\_interest, num\_month\_payments, total\_interest);  compute\_monthly\_payment(loan, total\_interest, num\_month\_payments, monthly\_payment);  ...........  } |

## Friday Group

The EPF and SOCSO rates for both employees and employers are constants. They can be defined in two possible ways: both are acceptable.

|  |
| --- |
| #define SOCSO\_EMPLOYEE 14.75  #define SOCSO\_EMPLOYER 51.56  #define EPF\_EMPLOYEE 0.11  #define EPF\_EMPLOYER 0.13  int main(void)  {  .....  .....  OR  int main(void)  {  const double SOCSO\_EMPLOYEE = 14.75  const double SOCSO\_EMPLOYER = 51.56  const double EPF\_EMPLOYEE = 0.11  const double EPF\_EMPLOYER = 0.13  ....  ....  } |

Functions 1, 2 and 3 may or may not use addresses / references. However, if the function is only returning a single value (as is the case here for all 3 functions) it is better to return the value directly rather than use an address / reference.

|  |
| --- |
| **// Example of calling functions 1 and 2 where the result is returned**  **// directly. This is the preferred approach**  double read\_salary();  double calculate\_epf(double gross\_pay, double rate);  ....  ....  int main(void)  {  double gross\_pay;  double epf\_employee\_deduct, epf\_employer\_deduct;  gross\_pay = read\_salary();  epf\_employee\_deduct = calculate\_epf(gross\_pay, EPF\_EMPLOYEE);  epf\_employer\_deduct = calculate\_epf(gross\_pay, EPF\_EMPLOYER);  .....  .....  }  **// Example of calling functions 1 and 2 where the result is**  **// returned via a reference**  void read\_salary(double& gross\_pay);  void calculate\_epf(double gross\_pay, double rate, double &deduct);  ...  ...  int main(void)  {    double gross\_pay = 0.0;  double epf\_employee\_deduct = 0.0;  double epf\_employer\_deduct = 0.0;    read\_salary(gross\_pay);    calculate\_epf(gross\_pay, EPF\_EMPLOYEE, epf\_employee\_deduct);  calculate\_epf(gross\_pay, EPF\_EMPLOYER, epf\_employer\_deduct);  } |