**Practical 3 – Refer to Topics 05 and 06**

**Part A (Understanding Concepts)**

1. Define the range of the random numbers generated by the following expressions:
2. rand( ) % 6 //n1 in range 0 to 5
3. rand( ) % 6 + 1 //n1 in range 1 to 6
4. rand ( ) % 6 + 11 //n3 in range 11 to 16
5. Which line(s) in the program below contain the following?
   1. A function prototype.
   2. A function call.
   3. A function definition (exclude the function main)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | #include <iostream>  using namespace std;  void instruct(void); //this is function prototype  int main(void)  {  instruct(); //function call  return 0;  }  void instruct(void) //line 12-15 function definition  {  cout << "This function displays program instructions.\n";  } |

1. Consider the following program.
2. What are the names of the functions in the program?

Main,func1,fun2

1. What does the reserved word ‘void’ in the parentheses in lines 4, 19 and 7 indicate?  
   the function requires no parameters
2. Identify any formal parameters or actual parameters used.

Forml – num in line 5 n 24

1. What is the output of the program?

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | #include <iostream>  using namespace std;  void func1(void);  void func2(int num);  int main(void)  {  int number;  number = -6789;  func1();  func2(123);  cout << "xyz";  func2(number);  return 0;  }  void func1(void)  {  cout << "ABC";  }  void func2(int num)  {  cout << num << endl;  } |

output: ABC123

xyz-6789

1. Determine if the following is a function prototype or a function call.

(a) display\_result(x + y); call

(b) void display\_result(int number); prototype

(c) void print\_line(void); prototype

(d) print\_line(); call

1. Given the function names below, draw a structure chart for a program that draws each of the following figures:

* main
* draw\_intersect
* draw\_horizontal
* draw\_parallel
* draw\_triangle

(a) (b)

**Part B (Programming Exercises)**

1. Run the following program and observe the output.

|  |
| --- |
| #include <iostream>  #include <iomanip>  #include <cmath> // for Math library functions  #include <cstdlib> // for random number generation functions  using namespace std;  int main(void)  {  double number1, number2, power;  number1 = 9.0;  number2 = 1.23;  power = 3;  cout << "Some Math library functions\n";  cout << fixed << setprecision(2);  cout << "sqrt(" << number1 << ")\t = " << setw(6) << sqrt(number1) << endl;  cout << "ceil(" << number2 << ")\t = " << setw(6) << ceil(number2) << endl;  cout << "floor(" << number2 << ")\t = " << setw(6) << floor(number2) << endl;  cout << "pow(" << number1 << "," << power << ")\t = "  << setw(6) << pow(number1, power) << endl;    cout << "\nRandom number generation functions\n";  srand(997);  cout << "srand(997) to set the seed\n";  cout << "rand(): " << rand() << endl;  cout << "rand(): " << rand()<< endl;  return 0;  } |

Output:  


1. Run the following program 2 times. Observe the random numbers generated.

|  |
| --- |
| #include <iostream>  #include <cstdlib> // for rand function  using namespace std;  int main(void)  {  int randNo[5];  for (int i = 0; i < 5; i++)  {  randNo[i] = rand() % 60 + 1; // to generate a random number  cout << randNo[i] << endl;  }  return 0;  } |

Output:  
42

48

35

41

30

Press any key to continue . . .

1. Run the following program 2 times. Observe the random numbers generated.

|  |
| --- |
| #include <iostream>  #include <cstdlib>  #include <ctime> // for time function  using namespace std;  int main(void)  {  int randNo[5];  // to generate different set of random numbers in each run  srand(time(NULL));  for (int i = 0; i < 5; i++)  {  randNo[i] = rand() % 60 + 1;  cout << randNo[i] << endl;  }  return 0;  } |

Random

1. Run the following program and observe the output.

|  |
| --- |
| #include <iostream>  using namespace std;  // Function prototypes  void draw\_horizontal(void);  void draw\_parallel(void);  int main(void)  {  draw\_horizontal();  draw\_parallel();  draw\_horizontal();  return 0;  }  /\*  \* Draws a horizontal line  \*/  void draw\_horizontal(void)  {  cout << "--------\n";  }  /\*  \* Draws parallel lines  \*/  void draw\_parallel(void)  {  cout << "| |\n";  cout << "| |\n";  cout << "| |\n";  } |

1. Write a program to draw the figures in Part A question 5 based on the structure charts you developed. You should reuse the functions in Part B question 4 wherever possible.
2. Complete the following program based on the comments given. The function displays the totals in the format shown below. The x’s represent values entered by the user or computed by the program. The price for an apple is RM0.85 and for an orange is RM1.00. The total is quantity (qty) multiplied by price. Note that the unit price of the fruits should be declared as constants using preprocessor directives.

Output Format:

=========My Fruit Store========

Fruit   Qty   Price    Total

   -------------------------------

   Apple    xx RM0.85   RMxx.xx

   Orange   xx RM1.00   RMxx.xx

   -------------------------------

   Total                  RMxx.xx

***Write or complete the statements at the positions marked with an arrow (->).***

|  |  |
| --- | --- |
| 1  2  3  **4->**  5  **6->**  7  8  9  10  **11->**  **12->**  **13->**  14  15  16  17  18  19  20  21  22  23  24  25  26  **27->**  **28->**  29  30  31  32  33  **34->**  **35->**  36  37  38  39  40  41  42  43  44  45  46  47  48  **49->**  50  51  52  53  **54->**  55  56  57  58  59  **60->**  61 | #include <iostream>  #include <iomanip>  // defined constant for price per apple  #define PRICE\_PER\_APPLE 0.85  // defined constant for price per orange  #define PRICE\_PER\_ORANGE 1.00  using namespace std;  // function prototypes  void display\_heading(void);  **void display\_apple\_item (int qty, double total);**  **void display\_orange\_item(int qty, double total);**  **void display\_grand\_total(double grand\_total);**  int main(void)  {  int apple\_qty, orange\_qty;  double apple\_total, orange\_total, grand\_total;  // get number of apples and oranges  cout << "Enter number of apples: ";  cin >> apple\_qty;  cout << "Enter number of oranges: ";  cin >> orange\_qty;  // compute total for apples and oranges  apple\_total = **apple\_qty \* PRICE\_PER\_APPLE;**  orange\_total = **orange\_qty \* PRICE\_PER\_ORANGE;**  // compute grand total  grand\_total = apple\_total + orange\_total;  // call the functions to display the result  display\_heading();  display\_apple\_item ( apple\_qty, apple\_total);  display\_orange\_item(**orange\_qty, orange\_total** ); display\_grand\_total(grand\_total);  return 0;  }  void display\_heading(void)  {  cout << "=========My Fruit Store========\n";  cout << "Fruit Qty Price Total\n";  cout << "-------------------------------\n";  }  void display\_apple\_item ( )  {  cout << **"Apple\t" << setw(3) << qty;**  cout << **" RM" << setw(4) << fixed << setprecision(2) << PRICE\_PER\_APPLE;**  cout << **"\tRM" << setw(5) << total << endl;**  }  void display\_orange\_item( )  {  cout << **"Orange\t" << setw(3) << qty;**  cout << **" RM" << setw(4) << fixed << setprecision(2) << PRICE\_PER\_ORANGE;**  cout << **"\tRM" << setw(5) << total << endl;**  }  void display\_grand\_total( )  {  cout << "-------------------------------\n";  **cout << "Total\t\t\tRM" << setw(5) << fixed << setprecision(2)**  << total << endl;  } |

**Part C (Self-Review / Revision)**

1. What is a function call?
2. What is a function definition?
3. What is a function prototype?
4. What is the purpose of a parameter?

Where do we write a formal parameter?

Where do we write an actual parameter?

**Part D (Practice Exercises)**

1. Write a program that displays the initials of your name. Use functions to display each letter. A sample run of the program is as follows (assuming the initials are HES):

H H

H H

HHHHH

H H

H H

EEEEE

E

EEEEE

E

EEEEE

SSSSS

S

SSSSS

S

SSSSS