**Practical 4 – refer to Topic 07 and 08**

**Part A (Understanding Concepts)**

1. Consider the following program that finds the average velocity of a particle travelling on a line between points p1 and p2 in time t1 to t2:
   1. How many parameters does each of the functions have and what are their types, if any?   
      main – none

Insctruct – none  
find\_velocity – 4 paremeters of type double

* 1. What is the return type of each of the functions?

Main-int  
instruct -none (void)

Find\_velocity -double

* 1. Is it valid to write this statement in function main? If no, explain why.

cout << "Average velocity is " << fixed << setprecision(2)

<< find\_velocity(4.5, 8.2, 0.0, 10.0) << endl;

Yes, valid

* 1. Is it valid to write this statement in function main? If no, explain why.

cout << instruct();

No, invalid because the function instruct() does not return any value to display using cout.

* 1. Write the program documentation for function find\_velocity using comments to describe its purpose and preconditions.

|  |  |
| --- | --- |
| 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 | #include <iostream>  #include <iomanip>  using namespace std;  void instruct(void);  double find\_velocity(double p1, double p2, double t1, double t2);  int main(void)  {  double ave\_velocity;  instruct();  ave\_velocity = find\_velocity(3.0, 9.0, 0.0, 2.0);  cout << "Average velocity is " << fixed << setprecision(2)  << find\_velocity(4.5, 8.2, 0.0, 10.0) << endl;  return 0;  }  double find\_velocity(double p1, double p2, double t1, double t2)  {  return (p2 - p1) / (t2 - t1);  }  void instruct(void)  {  cout << "This program calculates the average velocity\n";  cout << "of a particle travelling on a line between\n";  cout << "points p1 and p2 in time t1 to t2\n";  } |

/\*

\* Computes the average velocity of a particle travelling on a line

\* between points p1, and p2 in time t1 to t2.

\* Pre: p1, p2, t1, and t2 are defined.

\* p1 < p2 and t1 < t2.

\*/

1. Correct the errors in the following function definitions.
2. int func1(int x)

{

cout << "x = " << x << endl;

}

**void** func1(int x);

1. int func2(int x, y)

{

return x + y;

}

int func2(int x, **int** y)

1. Given the following function definition, what are the errors in the function calls below?

|  |
| --- |
| double func(double x, int y)  {  return x \* y / 100;  } |

1. z = func(2.0, 3, 100);

only 2 actual parameter allowed since function definition has only 2 formal parameters

1. z = func(double x, int y);

Should be func(x, y)

1. Consider the following program:
2. Identify the names of the local variables used in the functions.  
   In main: a, b In funcA: a In funcB: a, r
3. How it is possible for functions main, funcA and funcB to have a variable named a?  
   The concept of scope is applied.

The variable a in function main is only visible from the its declaration to the end of the function.

Similarly, the variables a in function funcA and funcB are only visible from its declaration to the end of the function.

1. Show the values of the variables and formal parameter for the program by filling the table below.

|  |  |
| --- | --- |
| 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 | #include <iostream>  using namespace std;  int funcA(void);  int funcB(int x);  int main(void)  {  int a, b;  a = funcA();  b = funcB(a);  cout << a << b << endl;  return 0;  }  int funcA(void)  {  int a;  cout << "Enter a: ";  cin >> a; // assume the value 23 is entered  return a;  }  int funcB(int x)  {  int a, r;  a = x / 10;  r = x - a \* 10;  return r;  } |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Statement executed** | **Data Area for function** | | |
| **main** | **funcA** | **funcB** |
| 1. | In main: funcA() | a b  ?  ? | a  ? |  |
| 2. | In funcA: cin >> a; | a b  ?  ? | a  23 |  |
| 3. | In funcA: return a;  In main: a = funcA(); | a b  ?  23 |  |  |
| 4. | In main: funcB(a) | a b  ?  23 |  | x a r  23  ?  ? |
| 5. | In funcB: a = x / 10; | a b  ?  23 |  | x a r  23  2  ? |
| 6. | In funcB: r = x – a \* 10; | a b  ?  23 |  | x a r  23  2  3 |
| 7. | In funcB: return r;  In main: b = funcB(a); | a b  ?  23 |  |  |

1. Consider the following program. Show the values of the variables and the output by filling the table below.

|  |  |
| --- | --- |
| 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 | #include <iostream>  using namespace std;  int strange (int x, int y);  int main()  {  int a, b, c, d, r, s;  a = 1;  b = 2;  c = 3;  d = 4;  r = strange (a, b);  cout << "r = " << r << endl;  s = strange (r, strange(c, d));  cout << "s = " << s << endl;  cout << "result = " << strange (r, s) << endl;  return 0;  }  int strange (int x, int y)  {  return x + y;  } |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Statement executed** | **Variables** | | | | | | **Output** |
| a | b | c | d | r | s |
| 1. | a=1;  b=2;  c=3;  d=4; | 1 | 2 | 3 | 4 | ? | ? | R=3  S=10  Result=13 |
| 2. | r = strange (a, b); | 1 | 2 | 3 | 4 | 3 | ? |
| 3. | cout << "r = " << r << endl; | 1 | 2 | 3 | 4 | 3 | ? |
| 4. | s = strange (r, strange(c, d)); | 1 | 2 | 3 | 4 | 3 | 10 |
| 5. | cout << "s = " << s << endl; | 1 | 2 | 3 | 4 | 3 | 10 |
| 6. | cout << "result = " << strange (r, s) << endl; | 1 | 2 | 3 | 4 | 3 | 10 |

1. Draw a structure chart with data flows for a program that has the following functions:

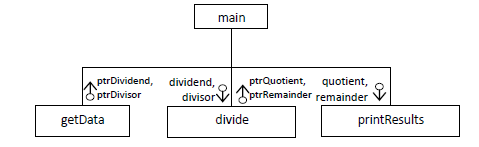
main – contains variables dividend, divisor, quotient, and remainder and calls

the 3 functions below.

getData – reads the values for the dividend and divisor.

divide – computes the quotient and remainder of the dividend divided by the divisor.

printResults – displays the quotient and the remainder.



**Part B (Programming Exercises)**

1. Rewrite the function main in Part A question 1 to ask the user for the values of p1, p2, t1, and t2 and then pass these values to function find\_velocity.
2. #include <iostream>
3. #include <iomanip>
4. using namespace std;
5. double find\_velocity(double p1, double p2, double t1, double t2);
6. int main(void)
7. {
8. double p1, p2, t1, t2;
9. cout << "Enter point1 and point2: ";
10. cin >> p1 >> p2;
11. cout << "Enter time1 and time2: ";
12. cin >> t1 >> t2;
13. cout << "Average velocity is " << fixed << setprecision(2)
14. << find\_velocity(p1, p2, t1, t2) << endl;
15. return 0;
16. }
17. double find\_velocity(double p1, double p2, double t1, double t2)
18. {
19. return (p2 - p1) / (t2 - t1);
20. }
21. Run the two programs below and observe the output. The first program uses the parameter passing method **pass-by-value** and the second program uses **pass-by-address**. Also note data type **short** is used.

|  |  |
| --- | --- |
| #include <iostream>  using namespace std;  void exchange(short x, short y);  int main(void)  {  short a, b;  a = 5;  b = 3;  exchange(a, b); // by value  cout << "a = " << a  << ", b = " << b << endl;  return 0;  }  void exchange(short x, short y)  {  short temp;  temp = x;  x = y;  y = temp;  return;  } | #include <iostream>  using namespace std;  void exchange(short\* x, short\* y);  int main(void)  {  short a, b;  a = 5;  b = 3;  exchange(&a, &b); // by address  cout << "a = " << a  << ", b = " << b << endl;  return 0;  }  void exchange(short\* x, short\* y)  {  short temp;  temp = \*x;  \*x = \*y;  \*y = temp;  return;  } |

1. Implement the program described in Part A question 6.

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

1. How do you pass information to a function when it is called?
2. How does a function give backs its result?
3. What is the general format for a function header?
4. What are the rules for actual and formal parameter list correspondence?
5. When is the data area for a function created? When is the data area destroyed?
6. What is the purpose of the concept of scope? What is the scope of a variable or parameter declared in a function? What is the scope of a function prototype?
7. What are the two parameter-passing methods used in C/C++ programs? What are the differences between the two methods?
8. Suppose a function reads two values from the user. How can the function pass back these two values to the calling function?
9. What is the \* symbol used for in a C/C++ program?

**Part D (Practice Exercises)**

1. (a) Write a function that returns the first digit in a given 4-dgit number. For example, given the integer 4537, the function returns 4.

(b) Write a main function to test the function you wrote.

1. The distance that a car (undergoing constant acceleration) will travel is given by the expression below where S=distance traveled, V=initial velocity, t=time travelled, and a=acceleration.



1. Write a function that reads value for initial velocity, time travelled, and acceleration.
2. Write a function that computes the distance traveled where the values of V, t and a are the parameters.
3. Write a main function to test the function you wrote. It calls the function to ask the user for values of V, t, and a, calls the function to compute the distance travelled and displays the result.