**Object Oriented Programming**

**Lab 1**

**Submitted To:**

Ma’am Amber Madeeha Zeb

**Submitted By:**

Manaal Waseem

FA18-BCE-074

**In Lab:**

**Task 1:**

**Write a program that declares a structure to store book Id, price and pages of a book. The structure should include functions to assign user defined values to each book and display the record of most costly book.**

**Code:**

1 #include <iostream>

2

3 #define size 5

4

5 **using namespace std**;

6

7 **struct** book

8 {

9 **int** id;

10 **int** price;

11 **int** pages;

12 };

13 **struct** book\_collection

14 {

15 **struct** book b[**size**];

16 **int** book\_id;

17 **int** price;

18 **int** pages;

19

20 **void** input\_data()

21 {

22 **for**(**int** a=0; a<=**size**; a++)

23 {

24 **cout** << "Enter Book Id: " << **endl**;

25 **cin** >> book\_id;

26 b[a].id = book\_id;

27

28 **cout** << "Enter Price: " << **endl**;

29 **cin** >> price;

30 b[a].price = price;

31

32 **cout** << "Enter pages: " << **endl**;

33 **cin** >> pages;

34 b[a].pages = pages;

35 }

36 }

37 **void** costly\_book()

38 {

39 **int** max\_val = 0;

40 **int** max\_idx = 0;

41

42 **for**(**int** j=0; j<**size**; j++)

43 {

44 **if**(b[j].price > max\_val)

45 {

46 max\_val = b[j].price;

47 max\_idx = j;

48 }

49 }

50 **cout** << "Costly Book: " << b[max\_idx].id << **endl**;

51 }

52 };

53

54 **int** main()

55 {

56 **struct** book\_collection bc;

57 bc.input\_data();

58 bc.costly\_book();

59

60 **return** 0;

61 }

**Output:**



**Task 2:**

**Write a program to take the values of two integers and use pointers to add 10 to the value of each integer.**

**Code:**

1 #include <iostream>

2

3 **using namespace std**;

4

5 **void** add\_ten(**int**& var1, **int**& var2) //formal parameters

6 {

7 var1\*=10; //change value of both actual and formal parameter

8 var2\*=10;

9 ;

10 }

11 **int** main ( )

12 {

13 **int** x, y;

14

15 **cout** << "Enter two integers: " << **endl**;

16 **cin** >> x >> y;

17

18 add\_ten(x, y); // pass actual parameters to function

19

20 **cout** << "x=" << x << ", y=" << y;

21

22 **return** 0;

23 }

24

**Output:**

****

**Task 3:**

**Write a function that swaps the values of two integer variables:**

**a. using pass by value**

**b. and pass by reference and see their differences**

**Code:**

1 #include <iostream>

2

3 **using namespace std**;

4

5 **void** swap\_value(**int** a, **int** b);

6 **void** swap\_reference(**int**& a, **int**& b);

7

8 **int** main()

9 {

10 **int** n1, n2;

11

12 **cout** << "Enter two numbers: " << **endl**;

13 **cin** >> n1 >> n2;

14

15 **cout** << "Swapped by value: " << **endl**;

16 swap\_value(n1,n2);

17

18 **cout** << "Swapped by reference: " << **endl**;

19 swap\_reference(n1,n2);

20 **cout** << "x = " << n1 << '\t' << "y = " << n2 <<**endl**;

21

22 **return** 0;

23 }

24

25 **void** swap\_value(**int** a, **int** b)

26 {

27 **int** temp;

28

29 temp=a;

30 a=b;

31 b=temp;

32

33 **cout** << "Numbers after swapping: " << **endl**;

34 **cout** << "x = " << a << '\t' << "y = " << b <<**endl**;

35 }

36 **void** swap\_reference(**int**& a, **int**& b)

37 {

38 **int** temp;

39

40 temp=a;

41 a=b;

42 b=temp;

43 }

**Output:**

****

**Post Lab:**

**Task 1:**

**There is a structure called employee that holds information like employee code, name, date of joining. Write a program to create an array of the structure and enter some data into it. Then ask the user to enter current date. Display the names of those employees whose tenure is 3 or more than 3 years according to the given current date.**

**Code:**

1 #include <iostream>

2 #include<conio.h>

3 #include <string>

4

5 **using namespace std**;

6

7 **struct** employee

8 {

9 **int** employee\_code;

10 **string** name;

11 **int** date;

12 **int** month;

13 **int** year;

14 };

15

16 **int** main()

17 {

18 **int** date;

19 **int** month;

20 **int** year;

21

22 **struct** employee e[2];

23

24 **cout**<<"Enter employee code, name, date of joining of a employee: "<<**endl**;

25

26 **for**(**int** i=0; i<2; i++)

27 {

28 **cout**<<"\nEnter details for employee: "<<**endl**;

29

30 **cout**<<"\nEnter employee code:";

31 **cin**>> e[i].employee\_code;

32

33 **cout**<<"\nEnter name:";

34 **cin**>>e[i].name;

35 **cin**.ignore(1,'\n');

36

37 **cout**<<"\nEnter date of joining (dd/mm/yyyy):";

38 **cin**>>e[i].date>> e[i].month>> e[i].year;

39 **cout**<< " d "<<e[i].date<<" m "<< e[i].month<< " y " << e[i].year<<**endl**;

40

41 }

42

43 **cout**<<"\nEnter current date (dd/mm/yyyy):";

44 **cin**>>date>> month>> year;

45 **cout**<< " d "<<date<<" m "<< month<< " y " << year<<**endl**;

46

47 **cout**<<"\nNames of employees whose tenure is 3 or more than 3 yrs:\n";

48

49 **for**(**int** j=0; j<2; j++)

50 {

51 **if**((year-e[j].year)>3)

52 **cout**<<e[j].name<<**endl**;

53 **else**

54 **if**((year-e[j].year)==3)

55 {

56 **if**((month-e[j].month)>=0)

57 **cout**<<e[j].name<<**endl**;

58 **else**

59 **if**((date-e[j].date)>=0)

60 **cout**<<e[j].name<<**endl**;

61 }

62

63 }

64 **return** 0;

65 }

66

**Output:**



**Task 2:**

**Write a function to sort data (in increasing order) in an array using**

**a. pass by value**

**b. and pass by reference.**

**Code:**

1 #include <iostream>

2 #include <stdlib.h>

3

4 #define ARRAY\_SIZE 50

5

6 **using namespace std**;

7

8 **int** find\_max(**int** \* ptr\_array, **int size**);

9 **void** selection\_sort\_reference(**int** \* ptr\_array, **int size**);

10 **void** selection\_sort\_value(**int** arr[ARRAY\_SIZE]);

11

12 **int** main()

13 {

14

15 **int** num\_array[ARRAY\_SIZE]; **/// Declare an integer array**

16

17 **int** \* ptr\_ar = &num\_array[0]; **/// A pointer to the start of the array**

18

19 **for**(**int** i =0; i < ARRAY\_SIZE; i++)

20 {

21 num\_array[i] = rand()%100; **/// Initialize the array with random numbers in range 0 to 99**

22 **cout** << num\_array[i] << "\t"; **/// and print it.**

23 }

24

25 **cout** << **endl**;

26

27 **int** mx\_idx = find\_max(ptr\_ar, ARRAY\_SIZE); **/// Print the maximum value and its index**

28 **cout** << "The maximum number is " << num\_array[mx\_idx] << "at index" << mx\_idx << **endl**;

29

30 **cout** << "Sort by reference: " << **endl**;

31 selection\_sort\_reference(num\_array, ARRAY\_SIZE); **/// Sort the array using Selection Sort**

32

33

34 **for**(**int** i=0; i<ARRAY\_SIZE; i++) **/// Print the sorted array**

35

36 **cout** << num\_array[i] << "\t";

37 **cout** << **endl**;

38

39 **cout** << "Sort by value: " << **endl**;

40

41 selection\_sort\_value(num\_array);

42

43 **return** 0;

44 }

45

46 **int** find\_max(**int** \* ptr\_array, **int size**)

47 {

48 **int** max\_val = 0;

49 **int** max\_idx = 0;

50

51 **for**(**int** i=0; i<**size**; i++)

52 {

53

54 **if**(\*(ptr\_array+i) > max\_val)

55 {

56 max\_val = \*(ptr\_array+i);

57 max\_idx = i;

58 }

59 }

60

61 **return**(max\_idx);

62

63 }

64

65 **void** selection\_sort\_reference(**int** \* ptr\_array, **int size**)

66 {

67 **int** temp, **max**, j;

68

69 **for**(j=**size**-1; j >0; j--)

70 {

71 **max** = find\_max(ptr\_array, j+1);

72 temp = \*(ptr\_array+**max**);

73 \*(ptr\_array+**max**) = \*(ptr\_array+j);

74 \*(ptr\_array+j) = temp;

75

76 }

77 }

78

79 **void** selection\_sort\_value(**int** arr[ARRAY\_SIZE])

80 {

81 **int** temp, **max**, j;

82 **int size** = ARRAY\_SIZE;

83 **for**(j=**size**-1; j >0; j--)

84 {

85 **max** = find\_max(&arr[0], j+1);

86 temp = arr[**max**];

87 arr[**max**] = arr[j];

88 arr[j] = temp;

89 //cout << arr[j] << "\t";

90 }

91

92 for(j=0; j < size ; j++)

93 {

94 cout << arr[j] << "\t";

95 }

96

97 }

**Output:**

****

**Task 3:**

**Write a program that inputs a string value from the user and displays it in reverse using pointer.**

**Code:**

1 #include <iostream>

2

3 **using namespace std**;

4

5 **int** main()

6 {

7 **string** str;

8

9 **cout**<< "Enter string: ";

10 **cin**>> str;

11 **int** len = str.length();

12 **int** n=len-1;

13

14 **for**(**int** i=0;i<(len/2);i++)

15 {

16 **char** temp = str[i];

17 str[i] = str[n];

18 str[n] = temp;

19 n = n-1;

20 }

21

22 **cout** << str << **endl**;

23 }

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**THE END**