

Assignment 13: Function Overloading.

Function overloading is a kind of polymorphism in which there can be several functions with same name but different set of parameters, all functions conceptually carrying out the same task but there is variation depending on the arguments. The definition of functions should differ from each other by type and /or number of arguments in the argument list. One commonly found overloaded function in a Class is the constructor itself as there can be more than one possible ways of constructing an object.

Set A

- a) Implement a class 'printdata' with three member functions all with the same name 'print'

void print(int) - outputs value - <int>, that is, value followed by the value of the integer

void print (int, int) – outputs value – [<int>, <int>], that is, value followed by the two integers separated by comma in square brackets.

void print(char *) – outputs value –"char*", that is, value followed by the string in double quotes.

Write a main function that uses the above class and its member functions.

- b) Implement a class 'maxdata' with two member functions both with the same name 'maximum'

int maximum (int, int) – returns the maximum between the two integer arguments

int maximum (int *) – returns the maximum integer in the array of integers

- Write a main function that uses the above class and its member functions.

Set B

- a) Implement a class 'invertdata' with three member functions all with the same name 'invert'

int invert (int) - returns the inverted integer – invert(5438) will return 8345

char * invert (char *) – returns the reversed string – reverse("comp") will return "pmoc"

void invert(int *) – will reverse the array order – An array [5, 7, 12, 4] will be inverted to

[4, 12, 7, 5]

Set C

- a) Can we have two functions with same name and set of arguments but different return data types?

b) Can we have two functions with same name with two different data types, for one function the argument is int for other it is user defined type 'Number' which is basically int but renamed 'Number' using typedef?

c) Can we have two functions void f(int x) and void f(int & x), that is, one with integer argument and other with reference to integer argument?

d) Can we have two functions given as follows?

```
void value(float x) { cout << "float " << x << endl; }
```

```
void value(double x) { cout << " double" << x << endl; }
```

What will be printed for the call

```
value( 10);
```

```
value (10.1);
```

e) Can we have two functions with same name, one having integer as argument while other enumerated data type as argument?

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: WellDone []

Assignment 3: Searching Techniques

The commonly used searching methods used are linear search and Binary search

Linear search is very simple technique to be used on any file while Binary search requires the file to be sorted.

Set A

- a) Create a random array of n integers. Accept a value x from user and use linear search algorithm to check whether the number is present in the array or not and output the position if the number is present.
- b) Create a random array of n integers. Sort the array using bubble sort. Accept a value x from user and use binary search algorithm to check whether the number is present in array or not and output the position if the number is present.

Set B

- a) Read the data from file 'cities.txt' containing names of 100 cities and their STD codes. Accept a name of the city from user and use linear search algorithm to check whether the name is present in the file and output the STD code, otherwise output "city not in the list".
- b) Read the data from file 'sortedcities.txt' containing names of 100 cities and their STD codes. Accept a name of the city from user and use binary search algorithm to check whether the name is present in the file and output the STD code, otherwise output "city not in the list".

Set C

- a) If the file contains multiple occurrences of a given element, linear search will give the position of the first occurrence, what modifications are required to get the last occurrence?
- b) If the file contains multiple occurrences of a given element, will binary search output the position of first occurrence or last occurrence?
- c) Which is best case search when searching using linear search and when using binary search?
- d) What modifications are required to linear search and binary search algorithm to count the number of comparisons?
- e) What modifications are required to binary search so that it returns the position where x can be inserted in the sorted array to retain the sorted order?

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: WellDone []