

USER DEFINE FUNCTIONS

User Define Functions as a way to define operations/methods/messages. Pure functions return values and do not change state, impure functions may return values but also change state, return type argument to function, function prototype and function signature, overloading. Variable of a class type as reference to an objects, invocation of function on objects through the reference, the concept of this.

Argument passing in functions, pass by value, what happened when a reference is passed side effect.

Q. What is user Define Functions? Why do we use functions while programs handling?

Ans: A named unit of a group of programs statements. This unit can be invoked from other parts of the program.

Q. Define User Define Function prototype?

Ans: The function prototype is the first line of the function definition that tells the program about the type of the value returned by the function and the number and types of arguments.

Q. What is the use of void before function name?

Ans: void data type specifies an empty set of values and it is used as the return type for functions that do not return a value.

Thus a function that does not return a value is declared as follows. void <functions name> (parameter list)

Q. Explain Functions/Methods Definitions with syntax?

Ans: A function must be defined before it is used anywhere in the program.

[access specifier][modifier]return-type
function-name (parameter list)
{

body of the function

}

[access specifier] can be either Public, Protected or Private. [modifier] can be one of final, native, synchronize, transient, volatile. return-type specifies the type of value that the return statement of the function returns. It may be any valid Java data type. parameter list is comma separated list of variables of a function.

Q. Why main() function so special?

Ans: The main() function is invoked in the system by default. hence as soon as the command for execution of the program is used, control directly reaches the main() function.

Q. Explain the function prototype and the signature?

Ans: The function prototype is the first line of the function definitions, that tells the program about the type of the value returned by the function and the number

and type of the arguments. Function signature basically refers to the number and types of the arguments, it is the part of the prototype.

Q. Explain the function of a return statement?

Ans: The return statement is useful in two ways. First an immediately exit from the function is caused as soon as a return statement is encountered and the control back to the main caller. Second use of return statement is that it is used a value to the calling code.

Q. Write advantages of using functions in programs.

Ans: (i) functions lessen the complexity of programs (ii) functions hide the implementation details (iii) functions enhance reusability of code

Q. Difference between Actual argument and Formal argument? [2007,2008]

Ans: The parameter that appears in

function call statement are called actual argument and The parameter that appears in function definition are called formal parameter.

Q. What are static members?

Ans: The members that are declared static is called static members. These members are associated with the class itself rather than individual objects, the static members and static methods are often referred to as class variables and methods.

Q. What is the use of static in main() methods?

Ans: (i) They can only call other static methods. (ii) They can only access static data. (iii) They can not refer to this or super in any way.

Q. What is call by value?

Ans: (i) In call by value, the called functions creates its own work copy for the passed parameters and copies the passed values in it. Any changes that take place

remain in the work copy and the original data remains intact.

Q. Explain the term “passed by reference”?

Ans: In passed by reference, the called function receives the reference to the passed parameters and through this reference, it access the original data. Any changes that take place are reflected in the original data.

Q. Differentiate between call by value and call by reference?

Ans: In call by value, the called functions creates its own work copy for the passed parameters and copies the passed values in it. Any changes that take place remain in the work copy and the original data remains intact. In call by reference, the called function receives the reference to the passed parameters and through this reference, it access the original data. Any changes that take place are reflected in the original data.

Q. Define an impure functions?

Ans: Impure Function change the state of the object arguments they have received and then return. The following functions is the example of an impure function:

```
public static Time increment(Time obj,  
double secs)  
{  
time.seconds+=secs;  
return(Time);  
}
```

Q. What is the difference between pure and impure functions?

Ans: Pure Function: These functions takes objects as an arguments but does not modify the state of the objects. The result of the pure function is the return value.

Impure Function: These functions change the state of the object arguments they have received.

Q. How are following passed in Java?

(i) primitive types (ii) reference types

Ans: (i) By value, (ii) By reference.

Q. What does function overloading mean?

What is its significance?

Ans: A Function name having several definitions in the same scope that are differentiable by the number or type of their arguments, is said to be an overloaded function. Function overloading not only implements polymorphism but also reduce the number of comparisons in a program and there by makes the programs run faster.

Q. Illustrate the concept of function overloading with the help of an example.

Ans:- A function name having several definitions that are differentiable by the numbers or types of their arguments is known as function overloading. For example following code overloads a function area to computer areas of circle

```

rectangle and triangle.
float area (float radius)           //circle
{
return (3.14 * radius * radius);
}
float area (float length, float
breadth) //rectangle
{
return (length*breadth);
}
float area (float side1, float  side2, float
side3) //area of triangle
{
float s = (side1 + side2 + side3)/2;
float ar = Math.sqrt(s * (s- side1)*(s-side2)
*(s-side3));
return (ar);
}

```

Q. What is ‘this’ keyword? What is its significance?

Ans: The “this” keyword is used to refer to currently calling objects. The member functions of every objects have access to a

sort of magic keyword name this, which points to the object itself. Thus any member function can find out the address of the object of which it is a member. The 'this' keyword represents an object that invokes a member function. It stores the address of the object that invoking a member function and it is an implicit argument to the member function being invoked. The 'this' keyword is useful in returning the object of which the function is a member.

Q. What do you mean by recursive function?

Ans: When a method is called inside its own definition the process is known as functions recursion and this function called recursive function.

Q. What is the difference between Methods and Functions?

Ans: The major difference between methods and functions is that methods

called by the reference variables called objects where as the functions do not having any reference variables.