Function Overloading

Function Overloading

- We can design family of functions with one function name and different argument lists.
- Function performs different operations depending on argument list in function call.
- Correct function to be invoked is determined by checking the number and type of arguments.

Match the following:

Function prototypes:

1. int add(int a, int b);
2. int add (int a , int b, int c);
3. double add (double x, double y);
4. double add(int p, double q);
5. double add(double p, int q);

Function calls:

1. cout<<add(15, 10.0);
2. cout<<add(5, 10);
3. cout<<add(12.5, 7.5);
4. cout<<add(0.75, 5);
5. cout<<add(5, 10, 15);

- Function call first matches the prototype having same number and type of arguments and calls appropriate function for execution.

- Best match function selection is as follows:

  - Compiler first tries to find exact match - type of actual arguments are same.

  - If exact match not found, integral promotions to actual arguments are done to find a match :
    - such as char to int, float to double

- If both fails, compiler tries built-in conversions to the actual arguments

  long square (long n)
  double square (double x)

Function call: **square(10)** causes error because *int* argument can be converted to *long* or *double* but situation is ambiguous.

- All fails then user-defined conversions are attempted.

- Caution: Should not overload unrelated functions and should reserve this for closely related tasks.
- Default arguments may be used instead of overloading.

```
/*Ex.1. Function overloading. Function volume()is overloaded
3 times */
#include <iostream.h>
int volume(int); // cube volume
int volume(int, int, int); // box volume
float volume(int, int); // cylinder volume
const float pi = 3.14;

int main()
{
 cout << "Cube volume : " << volume(5) << "\n";
 cout << "Box volume : " << volume(9, 3, 4) << "\n";
 cout << "Cylinder volume : " << volume(5, 6) << "\n";
 return 0;
}
int volume(int a)
{  return (a*a*a); }

int volume(int l, int b, int h)
{  return (l * b * h); }

float volume(int r, int h)
{  return (pi * r * r * h); }
```

```
/*Ex.2 Function overloading.
Function abslt()is overloaded 3
times */
int abslt(int);
long abslt(long);
float abslt(float);
double abslt(double);
int main()
{
 int intgr=-5;
 long lng=34225;
 float flt=-5.56;
 double dbl=-45.6768;
 cout<<"absoulte value of "
     <<intgr<<"="<<abslt(intgr)
     <<endl;

cout<<" absoulte value of "
    <<lng<<"="<<abslt(lng)<<endl;

cout<<" absoulte value of"
   <<flt<<"="<<abslt(flt)<<endl;

cout<<" absoulte value of "
    <<dbl<<"="<<abslt(dbl)<<endl;
}
```

```
int abslt(int num)
{
 if(num>=0)return num;
 else return (-num);
}

long abslt(long num)
{
 if(num>=0)return num;
 else return (-num);
}

float abslt(float  num)
{
 if(num>=0)return num;
 else return (-num);
}

double abslt(double  num)
{
 if(num>=0)return num;
 else return (-num);
}
```