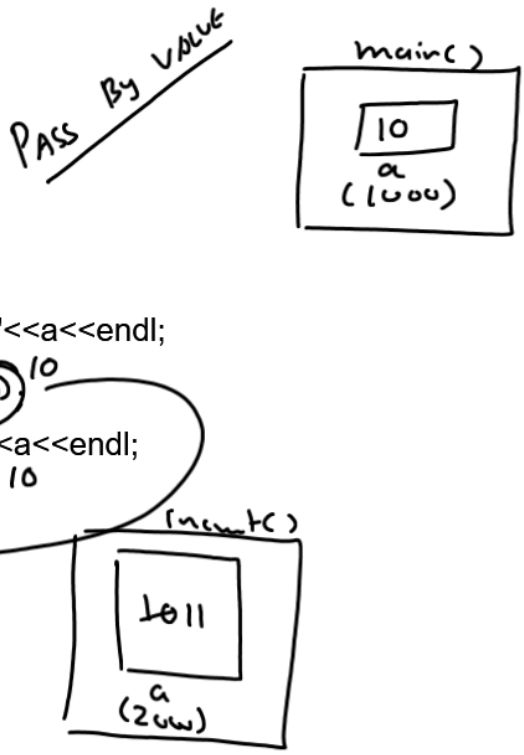
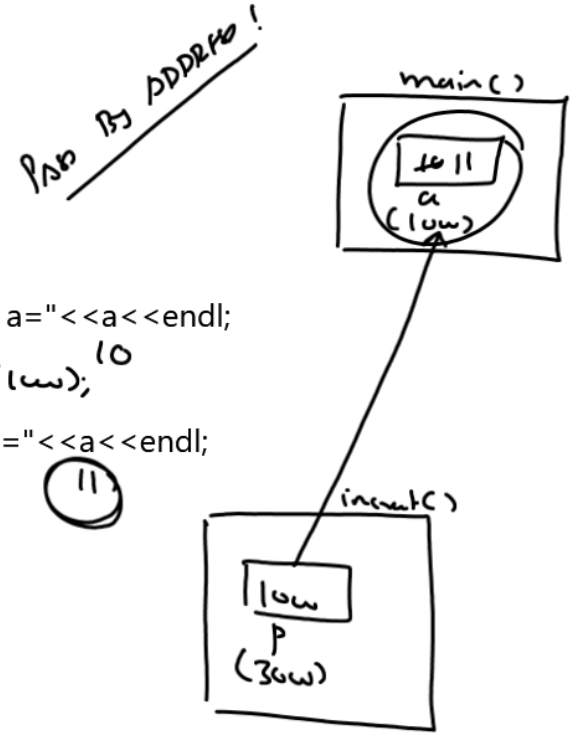
```
#include <iostream.h>
#include <conio.h>
void increment(int );
int main()
{
   int a=10;
   clrscr();
   cout<<"Before incrementing a="<<a<<endl;
   increment(a);
   cout<<"After incrementing a="<<a<<endl;
   return 0;
}
void increment(int a )
{
    a=a+1;
}
```




```
#include <iostream.h>
#include <conio.h>
void increment(int*);
int main()
{
   int a=10;
   clrscr();
   cout<<"Before incrementing a="<<a<<endl;
   increment(&a);
   cout<<"After incrementing a="<<a<<endl;
   return 0;
}
void increment(int *p)
{
   *p=*p+1;
}
```



```
#include <iostream.h>
#include <conio.h>
void increment(int &);
int main()
{
  int a=10;
  clrscr();
  cout<<"Before incrementing a="<<a<<endl;
  increment(a);
  cout<<"After incrementing a="<<a<<endl;
  return 0;
}
void increment(int &p)
{
   p=p+1;
}
```

Pass By Reference


```
class Student
{
    int roll;
    char grade;
    float per;
public:
    Student( );
    Student(int, char, float);
    Student(int, char);
    Student(Student &);
    Student( Student * );
    .
    .
    Student( int & );
};
```

Copy Const

Parameterized Const

Non Parameterized const

Parameterized Const

```
class Emp
{
   int age;
   char name[20];
   float sal;
public:
        Emp(Emp&); // COPY CONSTRUCTOR
        Emp();// NON PARAMETRIZED CONSTRUCTOR
        Emp(int,char *,float);//PARAMETRIZED CONSTRUCTOR
        Emp(int,float);// PARAMETRIZED CONSTRUCTOR
        Emp(char*,float);// PARAMETRIZED CONSTRUCTOR
        ....
};
```