```
#function classes objects realtime examples
```

```
# how to store a particular value and datatypes  and print
```

```
# function syntax def function name ():
# with attributes ,without attribute
#without attribute - greeting,welcome msg
def greeting():
  print("Hello welcome")
greeting()

#with attributes pin number,calculations,calculator
#it can take ip values and perform operations
def addition_fun(a,b):
  print("sum=",a+b)
  return a+b
addition_fun(10,20)
addition_fun(0,20)
addition_fun(10,2)
```

```
Hello welcome
sum= 30
sum= 20
sum= 12
12
```

```
def calculate_cost(item,qty,price):
  print("cost",qty*price)
  return
calculate_cost("banana",6,7)
calculate_cost("apple",60,10)
```

```
cost 42
cost 600
```

```
#calculator
print("simple calculator")
def add(x,y):
  return x+y
def sub(x,y):
  return x-y
def mul(x,y):
  return x*y
def div(x,y):
  return x/y
#example
x=10
y=5
#print
print("addition",add(x,y))
print("subtraction",sub(x,y))
print("multiplication",mul(x,y))
print("division",div(x,y))
```

```
simple calculator
addition 15
subtraction 5
multiplication 50
division 2.0
```

```
#condirtional statement
age = 15
if age > 18:
  print("Adult")
elif age < 18:
  print("Children")
else:
  print("not eligible")
```

```
Children
```

```
balance = 1000
#welcome msg
print("Welcome to SBI!!")

def check_balance():
    print("Your balance is: ", balance)
def deposit():
  global balance
  amount = float(input("Enter amount to deposit: "))
  balance += amount
  print("Deposited successfully!")
def withdraw():
  global balance
  amount = float(input("Enter amount to withdraw: "))
  if amount > balance:
    print("Insufficient balance!")
  else:
    balance -= amount
    print("Withdrawn successfully!")
#user input
while True:
  print("1. Check Balance")
  print("2. Deposit")
  print("3. Withdraw")
  print("4. Exit")
  choice = int(input("Enter your choice: (1-4) "))
  if choice == 1:
    check_balance()
  elif choice == 2:
    deposit()
  elif choice == 3:
    withdraw()
  elif choice == 4:
    print("Thank you for using SBI!")
    break
  else:
    print("Invalid choice!")
```

```
Welcome to SBI!!
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: (1-4) 1
Your balance is:  1000
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: (1-4) 2
Enter amount to deposit: 1000
Deposited successfully!
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: (1-4) 3
Enter amount to withdraw: 1000
Withdrawn successfully!
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: (1-4) 4
Thank you for using SBI!
```

```
#class is blueprint object is realtime entity
#class acting as container but object gives the memory
class Product:
  def __init__(self,name,price,qty=1) -> None:
    self.name=name
    self.price=price
    self.qty=qty
#shoping cart
class ShoppingCart:
  def __init__(self) -> None:
    self.products=[] #list of items
  def add_item(self,item):
    self.products.append(item)
    self.products.append(item)
  def display_items(self):
    total = 0
    print("Shopping cart content")
    for product in self.products:
      cost = product.price*product.qty
      print(product.name,product.price,product.qty,cost)
      total+=cost
    print("total cost",total)

prod1 = Product("Laptop",80000,1)
prod2 =Product("Headphone",50,2)

cart = ShoppingCart()
cart.add_item(prod1)
cart.add_item(prod2)
cart.display_items() #display
```

```
Shopping cart content
Laptop 80000 1 80000
Headphone 50 2 100
total cost 80100
```