```
### ![](https://ga-dash.s3.amazonaws.com/production/assets/logo-9f88ae6c9c3871690e33280fcf557f33.png)
Python Programming: Inheritance Exercise

<!---

Questions? Comments?
1. Log an issue to this repo to alert me of a problem.
2. Suggest an edit yourself by forking this repo, making edits, and submitting a pull request with
your changes back to our master branch.
3. Hit me up on Slack @susiremondi
--->

# Inheritance in Bank Accounts

## Overview:

You will practice writing classes and using inheritance by modeling different types of Bank accounts.

You will practice these programming concepts we've covered in class:
- Classes
- Inheritance

## Deliverables

One `.py` file with code that solves the problem.

## Requirements
You task is to write a series of classes that meet the criteria outlined below.

---

**Directions**

* Create a base **BankAccount** class
  * Bank accounts keep track of their current `balance`
  * Bank accounts have a `deposit` method
  * Bank accounts have a `withdraw` method
  * the `deposit` method returns the balance of the account after adding
    the deposited amount.
  * the `withdraw` method returns the amount of money that was successfully
    withdrawn.
  * Bank accounts return `False` if someone tries to deposit or withdraw
    a negative amount.
  * Bank accounts are created with a default interest rate of 2%
  * Bank accounts have a `accumulate_interest` method that sets the balance
    equal to the balance plus the balance times the interest rate
  * `accumulate_interest` returns the balance of the account after calculating
    the accumulated interest
* Create a **ChildrensAccount** class
  * Children's bank accounts have an interest rate of Zero.
  * Every time `accumulate_interest` is executed on a Child's account the
    account  always gets $10 added to the balance.
* Create an **OverdraftAccount** class
  * An overdraft account penalizes customers for trying to draw too much
    money out of their account.
  * Overdraft accounts are created with an `overdraft_penalty` property
    that defaults to $40.
  * Customer's aren't allowed to withdraw more money than they have in their
    account. If a customer tries to withdraw more than they have then the
    withdraw method returns `False` and their balance is deducted only by
    the amount of the `overdraft_penalty`.
  * Overdraft accounts don't accumulate interest if their balance is below zero.

**Sample Input:**: You can copy the below to test your code. The **sample output** below that is what
```

you should get.

```python
basic_account = BankAccount()
basic_account.deposit(600)
print("Basic account has ${}".format(basic_account.balance))
basic_account.withdraw(17)
print("Basic account has ${}".format(basic_account.balance))
basic_account.accumulate_interest()
print("Basic account has ${}".format(basic_account.balance))
print()

childs_account = ChildrensAccount()
childs_account.deposit(34)
print("Child's account has ${}".format(childs_account.balance))
childs_account.withdraw(17)
print("Child's account has ${}".format(childs_account.balance))
childs_account.accumulate_interest()
print("Child's account has ${}".format(childs_account.balance))
print()

overdraft_account = OverdraftAccount()
overdraft_account.deposit(12)
print("Overdraft account has ${}".format(overdraft_account.balance))
overdraft_account.withdraw(17)
print("Overdraft account has ${}".format(overdraft_account.balance))
overdraft_account.accumulate_interest()
print("Overdraft account has ${}".format(overdraft_account.balance))
```

Sample Output:
```
Basic account has $600
Basic account has $583
Basic account has $594.66

Child's account has $34
Child's account has $17
Child's account has $27

Overdraft account has $12
Overdraft account has $-28
Overdraft account has $-28
```